2002年3月1日発行（毎月1回1日発行）第21巻第3号 昭和57年12月6日第三種郵便物認可

わかる！動かせる！
プログラムが組める雑誌

マイコン BASIC Magazine

MAR. 2002
3
定価 550Yen



掲載プログラムがダウンロードできる！ http://www.basicmagazine.net/

平成14年度 大学入試センター試験出題 集中連載

数学①② 「BASIC言語」ズバリ解説

CPUの歴史と未来をさぐる！
1979年登場8ビットCPU 6809最強伝説編！

マンガで学ぶ！コンピュータの世界
忘れちゃいけない メモリ！

DOS/V実践自作講座
Linuxパソコンの組み立て！

- フリーソフトHSPで楽々3Dプログラミング！
- 図解！Visual Basicゲーム作成術
- はじめよう マイクロソフト ExcelでBASICプログラミング
- C言語でロボットを動かす

特選投稿プログラム・コーナー

● Visual Basic V6.0	チキチキバンバン2
● Visual Basic V6.0	HTMLテーブルメーカーゅ
● Visual Basic V6.0	Light
● Visual Basic V5.0	あーる・かるきゅれーしょんVer1.1
● F-BASIC V6.3	キツネとガチョウ
● HSP V2.5	サルも木から落ちないで
● HSP V2.5	Rolling Panels
● iアプリ	iエジプトの石
● iアプリ	ころころやかもてぃ
● PC-9801	バルーンタイプ
● FX-890P	熱帯魚



編集部の学校訪問記



各種プログラミング・コンテストで実績を上げる「情報技術研究部」
埼玉県立熊谷工業高等学校

▶PCとAV機器に対応！注目のソニー多機能PCディスプレイ

Empowered by Innovation　NEC

ADSLに対応。
バンドルータ⊿〈デルタ〉誕生。

のは、このスピード、高機能、そして快適性でした。



DR35FH

8MフルレートADSLモデム内蔵の本格派ブロードバンドルータ。これ1台で、高度なネットワーク利用にもフレキシブルに対応する、まさに本格派ADSLルータ。

●「8Mbps（G.dmt）／1.5Mbps（G.lite） AnnexC方式」対応のADSLモデムを内蔵。

⚠ 本商品でご利用いただけるADSL事業者およびインターネットプロバイダをホームページ「AtermStation」で必ずご確認ください。

●高度な設定も柔軟にカスタマイズできる高機能ブロードバンドルータ機能。

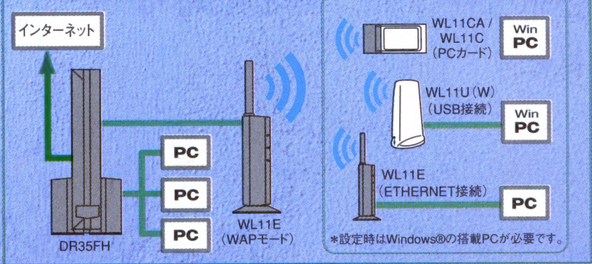
●外部公開用サーバを他と切り離しファイアウォールのDMZ領域化も可能。

●ビジネスユースにも対応する、豊富なルータ機能を満載。

・スタティックルーティング（最大40経路設定可能）・DHCPサーバ機能・ProxyDNS機能・IPパケットフィルタリング機能（IPアドレス／Macアドレス等による指定可）・IPマスカレード機能・ポートマッピング（ポート無変換機能付）・システムログ機能など

＊DIRECTSTAR⊿も「ウイルス警備隊V3」を添付CDに収録。

■WL11EをWAP（ワイヤレスアクセスポイント）として接続すれば、ワイヤレスLANでの利用も可能。



■4ポートスイッチングHUB搭載フルレート対応ADSLルータ

Aterm DR35FH/GS NEW　ACCA NETWORKS 推奨

WAN／ADSL回線接続:1　LAN／100BASE-TX/10BASE-T:4

Aterm DR35FH/CE NEW

WAN／ADSL回線接続:1　LAN／100BASE-TX/10BASE-T:4

■フルレート対応ADSLルータ

Aterm DR30F/GS NEW　ACCA NETWORKS 推奨

WAN／ADSL回線接続:1　LAN／10BASE-T:1

Aterm DR30F/CE NEW

WAN／ADSL回線接続:1　LAN／10BASE-T:1

掲載商品はすべてオープン価格です。

Aterm® DIRECTSTAR⊿

●「121ware.com」ホームページ http://121ware.com

商品に関するお問い合わせは

●Aterm（エーターム）インフォメーションセンター　0120-361138　TEL.0471-85-4761

受付時間:月～金曜日 午前9時～午後6時、土曜日 午前9時～午後5時（日曜日、祝祭日、年末年始、当社の休日はお休みさせていただきます。）

NECアクセステクニカ

マイコンBASIC Magazine

http://www.basicmagazine.net

3 2002 MARCH





© マイコンBASICマガジン3月号
第21巻 第3号 2002年2月25日印刷
2002年3月1日発行
定価 550円(本体524円)
電波新聞社 2002 Printed in Japan
編集発行人：平山哲雄
東京都品川区東五反田1-11-15
郵便番号 141-8715
電話　東京(03)3445-6111(大代表)
振替（東京）00150-3-51961
印刷：奥村印刷(株)　製本：(株)堅省堂



広告索引

編集部の学校訪問記


▲情報技術研究部のみなさん

埼玉県熊谷市の西に位置する県立熊谷工業高等学校（山下照夫校長）。前身は大正９年設立の熊谷商業学校で，昭和４１年県立熊谷商工高等学校から分離独立し，名門工業高校として着実に実績を上げてきています。

「文武両道」を校訓にしている同校は，ラグビー部が全国大会出場23回，優勝１回，準優勝１回の輝かしい成績をあげており，校名を耳にした人も多いと思いますが，運動部だけでなく，情報技術研究部も，プログラミングコンテストで最優秀賞を受賞するなどの成績を上げています。今回は，その情報技術研究部の活動と，情報技術科について取材してきました。

埼玉県立 熊谷工業高等学校

http://www.kumagaya-ths.ed.jp/

プログラミングコンテストや検定試験取得を目標とする「情報技術研究部」

全国で最初に設置された「情報技術科」

熊谷工業高校には現在，建築・土木・電気・機械・情報技術の五つの学科が設置されています。工業高校というと，卒業後の進路として就職というイメージを持つ人も多いでしょうが，同校は進学者が多く，大学進学：専門学校進学：就職＝1：1：1の比率というのも特色のひとつです。

情報技術科は，昭和46年に全国で最初に設置された学科で，実習に大きなウエイトを占め，ハードからソフトの最先端技術に対応できるエンジニアの育成を目指しています。①電気・電子を中心にしたハードウエア教育，②CやCOBOL，FORTRAN，JAVAなどのプログラミング言語教育，③ハード／ソフト製作などの実習教育を3本柱にしており，ソフトウェア開発技術者試験／基本情報技術者試験／初級システムアドミニストレータ試験などの国家試験にも数多く合格者を出しています。


▲メインフレーム機の富士通「GS8300/8G」

情報技術科の実習例

1年
工業基礎
電流・電圧計　オームの法則　テスタ製作　抵抗直並列　キルヒホッフ　パラレルポート制御　オシロスコープ　ダイオード
プログラミング技術
UNIXの基本操作　C言語によるプログラミング

2年
情報技術実習
FETの静特性　増幅回路の製作　増幅回路の特性　論理回路の基礎　カウンタ・シフトレジスタ　IC工作　光センサの製作　JCLの基礎　GUIプログラミング　インバータの製作　エンコーダ・デコーダ　マルチバイブレータの実習　UNIX実習
電子情報技術
FORTRAN言語によるプログラミング
ソフトウエア技術
アセンブラ言語（CASL2）

3年
情報技術実習
DELPHI実習　日本語フォームオーバレイ　オンラインシステム　ハードウエア入門　OA実習　JCL実習　PICマイコン
ソフトウエア技術
COBOL言語によるプログラミング
課題研究
ハード製作：ライントレースロボット　LEGOマインドストーム
ソフト製作：各種プログラム
選択プログラミング技術
C言語コース　オブジェクト指向技術（JAVA）

情報技術科の先生


▲科長の植原広之先生。「実習に重きを置いた教育を実践しています」


▲長島久先生。「昔，マイコンやベーマガを読んでましたよ」


▲山本哲也先生。「育ててもらったベーマガに協力できることは何でも」

博物館

廊下には「博物館」と称し，歴代使われてきた汎用コンピュータやディスク装置が展示されている

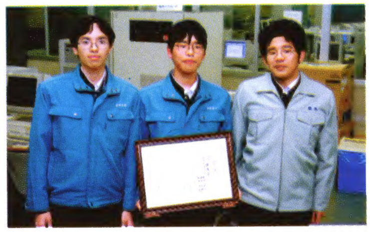



「目指せ全国プログラミングコンテスト最優秀賞!!」を合言葉に活動する

情報技術研究部

通称「S研」と呼ばれている情報技術研究部。現部長の権田真一くん（2年生）を中心に，Visual BasicやVisual C++などによるプログラミングを主に毎日活動しています。

1学期は新1年生の指導，2学期以降は各自それぞれ思い思いにプログラミングに励みます。目標は経済産業省主催の「全国プログラミングコンテスト」や埼玉県主催の「埼玉県工業高等学校プログラミングコンテスト」など，各種コンテストへの応募です。内容は，実用的なもの，ゲーム，ホームページなどさまざまです。



◀埼玉県プログラミングコンテスト・競技部門「優良賞」の賞状を持つ（左から）益田勇助くん，権田真一くん，鯨井裕之くん（それぞれ2年生）



▲情報技術科の「実習棟」にはパソコン室が2教室，それぞれ43台，計86台の富士通「FMV 6366DX2C」（OSはWindows98）が設置されている。今年の夏，パソコン更新予定



▲（左から）兼子重人くん，須田和宏くん，中島嘉男くん（3年生）は，数々のコンテストで最優秀や優秀賞などを受賞



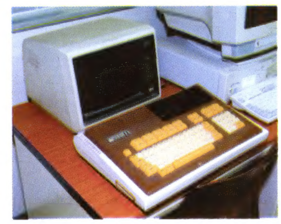
◀活動は各自それぞれ作成するものはちがえど，目指す目標はみな同じ。教えあったりなどのコミュニケーションもバッチリ！



◀埼玉県ロボットコンテストで見事優勝した兼子重人くんのマシン。「小泉学先生の指導のもと，ライントレースの基本アルゴリズムを一緒に考えた」という



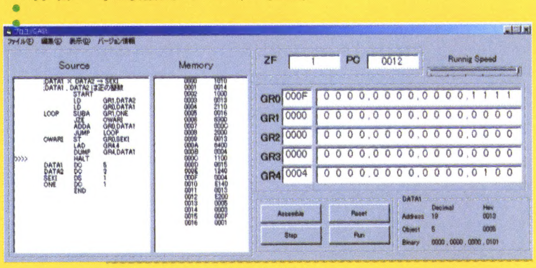
◀（左から）木村公亮くん、斎藤邦啓くん。1年生にして最優秀賞を受賞するレベルに達する生徒もいる



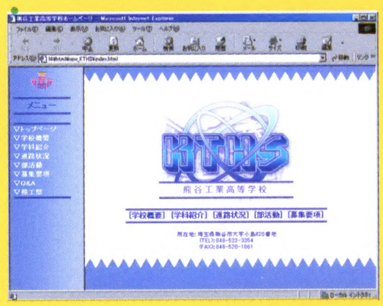
◀6809のシミュレータをハンドアセンブルで組んだりするためになつかしの「FM-8」も置いてあった

情報技術研究部　作品集

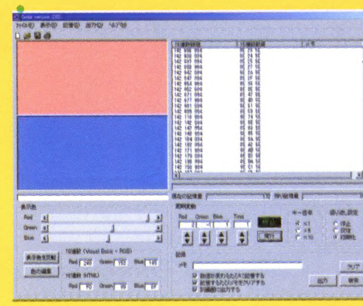
埼玉県工業高等学校プログラミングコンテスト
競技部門
最優秀賞　受賞
作品名：プロコンCOMET シミュレータ
作者：中島嘉男くん（3年生）



全国高校生専門学校生
プログラミングコンテスト
ホームページ部門　第3位受賞
作品名：埼玉県立熊谷工業高等学校ホームページ
作者：須田和宏くん（3年生）



全国情報技術研究会主催
プログラミングコンテスト
優良賞　受賞
作品名：Color Version 2.00
作者：木村公亮くん（1年生）



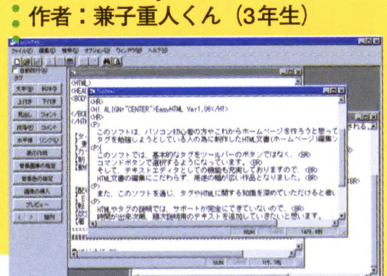




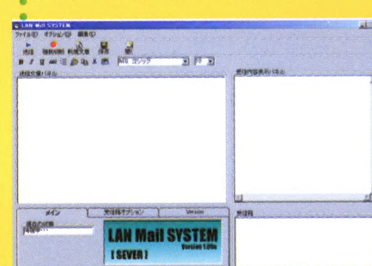
全国情報技術研究会主催
プログラミングコンテスト
佳作　受賞
作品名：ERA
作者：中島嘉男くん（3年生）



全国情報技術研究会主催
プログラミングコンテスト
優秀賞　受賞
作品名：EasyHTML
作者：兼子重人くん（3年生）



埼玉県工業高等学校
プログラミングコンテスト
プログラム部門
最優秀賞　受賞
作品名：LAN Mail SYSTEM Version 2.00a
作者：斎藤邦啓くん（1年生）



熊谷工業高等学校　〒360-0832 埼玉県熊谷市大字小島820番地　☎048-523-3354

game

ゲームで楽しく「歴史」を学ぶ!!

コーエー定番シリーズ紹介

コーエーから「定番シリーズ」として，過去の名作が手頃な価格（各1,980円）で続々と復刻されています。「古いゲームなので心配…」という人もいるかと思いますが，昨年末に発売されたゲームボーイアドバンス版「三國志」も，じつは「三國志Ⅳ」の復刻版なのです。新作のように楽しむことができる名作がラインアップされているので安心してください。

一方で，ゲームがきっかけで歴史に興味を持った人も数多くいます。歴史上のストーリーに参加することで，実際に自分がその場所にいるかのような楽しみを味わえるのが醍醐味なんです。

今回紹介する3作品は，RPGのように順を追ってシナリオを楽しむことができる「リコエイション・ゲーム」なので，時代背景や原典を知らない人や，SLGはちょっと苦手という人でも楽しめます。一度プレイしてみませんか？

学校の先生方には「歴史」の授業に使ってみるという新しい発想もお勧めします。

大航海時代Ⅱ

開拓初期の時代の探検，貿易を楽しむ!!

■Windows95/98/Me
■1,980円（税別）
■2月8日発売





15世紀から16世紀のヨーロッパは，大航海時代と呼ばれ，アメリカは新大陸，日本も黄金の国ジパングと呼ばれ，神秘のベールに包まれていました。ヨーロッパの各国は我先にと船を出し，世界中の大陸を往来していたのです。

シリーズ最高傑作の呼び声が高い「大航海時代Ⅱ」は6人の主人公が登場し，それぞれの人生を描いたストーリーを楽しむことができるものです。交易，海賊，冒険，地図作成などゲームの進めかたはプレイヤーしだいです。

太閤立志伝Ⅱ

立身出世の代名詞秀吉を天下人に導く!!

■Windows95/98/Me/XP
■1,980円（税別）
■2月8日発売





大河ドラマ「利家とまつ」で話題の戦国時代が舞台の「太閤立志伝Ⅱ」。

織田信長配下の足軽から天下人へと出世していく木下藤吉郎（秀吉）を主人公とする立身出世ゲームです。

歴史どおり信長に忠実に仕えて出世してもいいですし，命令を無視して全国行脚の旅に出ることも，信長に反旗を翻すこともできるなど，自由度もあります。安土桃山の文化を肌で感じながら，自分だけの秀吉像を作ってみませんか。

三國志孔明伝

壮大な三国志物語を諸葛孔明の視点で!!

■Windows95/98
■1,980円（税別）
■2月8日発売





中国2世紀末から3世紀にかけての乱世の時代を描いた「三国志演義」を再現。稀代の名政治家，名軍師として名高い諸葛孔明の生涯を楽しむことができます。

魅力ある登場武将は約200人。全40種類の舞台や戦場が「三国志演義」ならではの戦術や火計などの策略で彩られ，SLGの戦闘が苦手という人でも手軽に遊ぶことができます。

諸葛孔明の仕える蜀が中国統一をしていたら…という「もしも」を楽めるストーリー展開もあり，魏や呉のファンでもハラハラドキドキものです。

＋ここで紹介したソフトそれぞれ5本プレゼント！＋　＋応募方法は，126ページをご覧ください。＋

平成14年度
大学入試センター試験出題

☆☆☆ 集中連載 その3 ☆☆☆

数学①②「BASIC言語」ズバリ解説!!

これが正解だ! キミは出来たか?

東幸太

2002年1月19日に実施された平成14年度・大学入試センター試験の数学の問題には，選択問題としてBASICプログラムに関する問題が出題された。残念ながら先月号までの予想問題は的中というわけにはいかなかったが，解説してきたプログラム要素は随所に現れている。

今回は集中連載最終回として，出題された問題についての解法と解説を行いたいと思う。現役受験生の諸君は２次試験に向けて解法の確認をしてほしい。また，来年センター試験を受けることになる諸君は，最も好都合な例題としてこの問題に自力で挑戦した上で，解答との突き合わせをしてほしい。

＜数学Ⅰ・A＞

数学Ⅰ・A

第5問（選択問題）（配点20点）

次のプログラムは $x=0,1,\cdots\cdots,9$ に対する ax^2+bx+c の値の最小値と最大値を求めるものである。[アイウ]，[エオカ] に適当な行番号を入れてプログラムを完成せよ。

```
100 INPUT "a=";A
110 INPUT "b=";B
120 INPUT "c=";C
130 U=C
140 V=C
150 FOR X=0 TO 9
160     Y=A*X*X+B*X+C
170     IF Y>=U THEN GOTO [アイウ]
180     U=Y
190     IF Y<=V THEN GOTO [エオカ]
200     V=Y
210 NEXT X
220 PRINT "最小値=";U
230 PRINT "最大値=";V
240 END
```

(1)　上のプログラムを実行して，a=? に対して −1，b=? に対して 7，c=? に対して 28 を入力すると，180 行は [キ] 回，200 行は [ク] 回実行され

最小値 = [ケコ]
最大値 = [サシ]

が表示される。また，170 行の不等号 >= を > に，190 行の不等号 <= を < に変更したのち，同じデータを入力すると，180 行は [ス] 回，200 行は [セ] 回実行され

最小値 = [ソタ]
最大値 = [チツ]

が表示される。

(2)　冒頭のプログラムの 170 行と 180 行は，180 行を削除して 170 行を

170 [テ]

と書き直しても同じ結果を得る。同様に 190 行と 200 行も，200 行を削除して，190 行を

190 [ト]

と書き直すことができる。ただし，[テ] と [ト] については，次の⓪〜⑤の中から最もふさわしいものを一つずつ選べ。

⓪ IF Y>U THEN U=Y　　① IF Y<U THEN U=Y
② IF Y=U THEN U=Y　　③ IF Y>V THEN V=Y
④ IF Y<V THEN V=Y　　⑤ IF Y=V THEN V=Y

平成14年度大学入試センター試験本試験より

設問は２次式の最大値と最小値を求めるプログラムに関するものになっている。２次式の最大値・最小値といえば，微分した式の解を求めるのが定番だが，このプログラムでは評価の対象となる数値を順次２次式に当てはめていき，最大値と最小値を探すものになっている。これは，現実にも利用されるアルゴリズムで，極めて複雑な関数の最大値・最小値を求める際などに利用されている。

最初の設問は２か所の空欄に行番号を埋めてプログラムを完成させるというものである。プログラミング問題の場合，プログラムに求められている要素を問題文から正確に

読み取る必要がある。

問題文を見ると，

・ｘ＝０，１，・・・，９

・ax^2+bx+c

・最小値と最大値

といった要素が読み取れる。さらに，プログラムの150行からｘは０以上９以下の整数であること，ａ，ｂ，ｃは最初に入力された後，一定の定数として利用されていることがわかる。よって，「定数ａ，ｂ，ｃによる２次式ax^2+bx+cについて，ｘを０以上９以下の整数とした時の最小値と最大値を求める」ことが要求されていることがわかる。

プログラミング問題では，要求内容がはっきりした後に自分なりにプログラムを考えてみることが重要だが，今回の設問のように，空欄部分が限定的な場合は直接設問のプログラムを調べてみる方が手っ取り早い。

空欄になっているのは共に行番号であることから，処理自体はすでに示されている部分がすべてとなる。処理を調べていくと，以下のようになる。



≪第1図≫　数学Ⅰ・Aの問題のフローチャート

100行から120行	定数ａ，ｂ，ｃに値を設定している。
130行と140行	変数ｕとｖを設定している。これらの変数は220行と230行の表示部分からそれぞれ最小値と最大値に対応していることがわかる。
150行	変数ｘを０から９までループさせるためのFOR文である。
160行	設問の２次式の値を変数ｙに代入している。
170行	ｙとｕを比較し，ｙがｕ以上であればGOTO文が実行される。
180行	ｕにｙを代入している。ｙは２次式の値，ｕは最小値であるから，この部分で最小値の更新を行っていることがわかる。また，130行でｕをｃで初期化しているのは，ｃの値は最小値以上の値であることが保証されているためである。
190行	ｙとｖを比較し，ｖがｙ以上であればGOTO文が実行される。
200行	ｖにｙを代入している。ｙは２次式の値，ｖは最大値であるから，この部分で最大値の更新を行っていることがわかる。また，140行でｖをｃで初期化しているのは，ｃの値は最大値以下の値であることが保証されているためである。
210行	150行に対応するNEXT文である。
220行と230行	最小値と最大値を画面に出力している。

この調査結果から，このプログラムはｘを０から９まで順番にループさせていく中で，２次式の値ｙを計算し，その値と最小値候補，あるいは最大値候補と比較していることがわかる。この場合，最終的にｕに最小値，ｖに最大値が格納されるようにするためには，ｙの値が最大値候補より大きい場合や，最小値候補より小さい場合に，それぞれの候補の値をｙに変更していく必要がある。フローチャートとしては**第1図**のようになっている必要があるわけである。

170行のIF文が成立した場合というのは，最小値候補の更新が不要な場合である。よって，更新処理を行っている180行を実行しないようにする必要がある。このため，190行にジャンプする必要がある。また，同様に190行のIF文が成立した場合というのは，最大値候補の更新が不要な場合であり，ジャンプ先は200行を避けるための210行になる必要がある。

よって，**アイウは「１９０」，エオカは「２１０」**となる。

設問１では，ａを－１，ｂを７，ｃを２８に設定した場合の180行と200行の実行回数と，最終的に出力される最小値・最大値が求められている。

特定の行の実行回数が求められているためシミュレーションを避けることは難しい。ループ回数は10回なのでやや多いがループ内部の処理が簡単であるため，直接シミュレーションを行う。

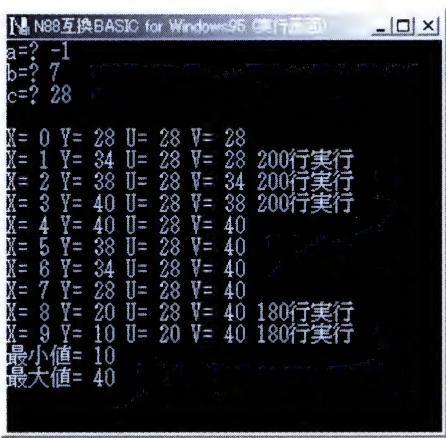
注目すべき変数はｘ，ｙ，ｕ，ｖであるため，160行の計算完了時のそれらの値の変化を追いかけていくと以下のようになる。

x	y	u	v	
0	２８	２８	２８	170行，190行共に成立するためｕとｖは変化しない
1	３４	２８	２８	190行は成立せず，ｖは200行で３４に更新される
2	３８	２８	３４	190行は成立せず，ｖは200行で３８に更新される
3	４０	２８	３８	190行は成立せず，ｖは200行で４０に更新される
4	４０	２８	４０	170行，190行共に成立するためｕとｖは変化しない
5	３８	２８	４０	170行，190行共に成立するためｕとｖは変化しない
6	３４	２８	４０	170行，190行共に成立するためｕとｖは変化しない
7	２８	２８	４０	170行，190行共に成立するためｕとｖは変化しない
8	２０	２８	４０	170行は成立せず，ｕは180行で２０に更新される
9	１０	２０	４０	170行は成立せず，ｕは180行で１０に更新される

結果として，180行は２回，200行は３回実行され，最小値は１０，最大値は４０となることがわかる。よって，**キは２，クは３，ケコは「10」，サシは「４０」**となる。問題のプログラムの表示を拡張した**リスト１**を実行した結果は**写真１**のようになり，この表と対応していることがわかる。

≪リスト1≫

```
100 INPUT "a=";A
110 INPUT "b=";B
120 INPUT "c=";C
130 U=C
140 V=C
150 FOR X=0 TO 9
160  Y=A*X*X+B*X+C
170  IF Y>=U THEN GOTO 190
180  U=Y
190  IF Y<=V THEN GOTO 210
200  V=Y
210 NEXT X
220 PRINT "最小値=";U
230 PRINT "最大値=";V
240 END
```



▲≪写真1≫　リスト1を実行した結果
問題のプログラムの１６０行実行時の状態を表示し，１８０行と２００行の実行状態を表示するように変更したリスト1の実行結果。UやVが更新されるべき時に１８０行や２００行が実行されている。

次に，170行と190行のIF文の条件式に含まれる等号「＝」を取り除いた場合についてであるが，この場合はｕやｖがｙと等しい場合はIF文が成立しなくなる。この点に注意して上記と同様のシミュレーションを行うと以下のようになる。

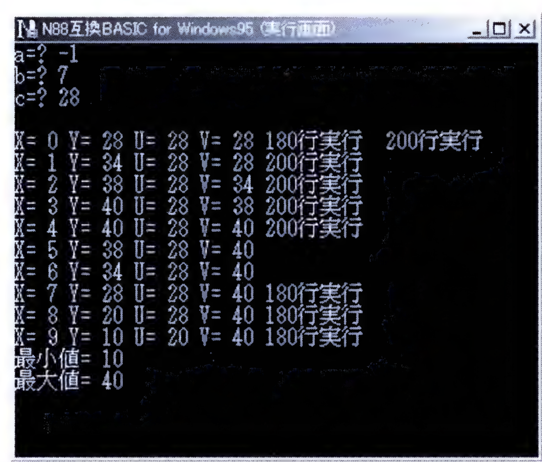
x	y	u	v	
0	２８	２８	２８	170行，190行共に成立せずｕは180行で，ｖは200行でそれぞれ２８に更新される
1	３４	２８	２８	190行は成立せず，ｖは200行で３４に更新される
2	３８	２８	３４	190行は成立せず，ｖは200行で３８に更新される
3	４０	２８	３８	190行は成立せず，ｖは200行で４０に更新される
4	４０	２８	４０	190行は成立せず，ｖは200行で４０に更新される
5	３８	２８	４０	170行，190行共に成立するためｕとｖは変化しない
6	３４	２８	４０	170行，190行共に成立するためｕとｖは変化しない
7	２８	２８	４０	170行は成立せず，ｕは180行で２８に更新される
8	２０	２８	４０	170行は成立せず，ｕは180行で２０に更新される
9	１０	２０	４０	170行は成立せず，ｕは180行で１０に更新される

違いが生じるのはｙとｕが等しい場合と，ｙとｖが等しい場合であり，ｘ＝０，４，７の時が該当する。この時にはｕとｖに対する代入処理自体は行われるものの，代入される値はｙと等しい値であり，ｙの値は変化せず，プログラムの出力結果に対しても影響はない。ただし，180行の実行回数は４回，200行の実行回数は５回にそれぞれ増加する。

よって，**スは４，セは５，ソタは１０，チツは４０**となる。170行と190行のIF文の条件式から等号を外した**リスト２**を実行すると**写真２**のようになり，この表と対応が取れていることがわかる。

≪リスト2≫

```
100 INPUT "a=";A
110 INPUT "b=";B
120 INPUT "c=";C
130 U=C
140 V=C
150 FOR X=0 TO 9
160  Y=A*X*X+B*X+C
165  PRINT : PRINT "X=";X;" Y=";Y;" U=";U;" V=";V;
170  IF Y>=U THEN GOTO 190
180  U=Y : PRINT " 180行実行 ";
190  IF Y<=V THEN GOTO 210
200  V=Y : PRINT " 200行実行 ";
210 NEXT X
215 PRINT
220 PRINT "最小値=";U
230 PRINT "最大値=";V
240 END
```

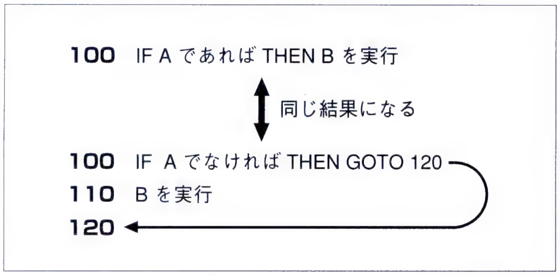


▲≪写真2≫　リスト2を実行した結果
リスト1のIF文の条件式に含まれる等号を取り除いたリスト2の実行結果。動作自体は変わらないものの，Yの値が最大値や最小値と等しい場合にも更新処理が行われるようになる。

設問２では，180行と200行をそれぞれ削除し，プログラムの出力結果に影響が出ないように170行と190行を選択肢の文に置き換えている。選

択肢の文は，IF文の後のTHEN以降に代入文が付けられたものになっている。

これは，第2図のようにIF文の記述方式を切り替える操作になる。THENの後にGOTO文を記述して処理をバイパスさせるパターンと，THENの後で直接処理を行うパターンの切り替えである。



≪第2図≫　IF文の記述のパターン

元のプログラムの170行と180行の組み合わせは，yが最小値になりうる場合の最小値の更新処理になっている。同様に，190行と200行の組み合わせは，yが最大値になりうる場合の最大値の更新処理になっている。このような処理を1行で実行できるものを選び出せばよい。

書き直した後の170行は最小値更新処理を行うためのu＝yの代入処理が，190行は最大値更新処理を行うためのv＝yの代入処理がそれぞれ必要になる。よって，対応する選択肢はそれぞれ三つになる。

最小値更新処理については，yがuよりも小さい場合にu＝yの処理を実行する必要がある。これに対応できるのは，1の「IF Y<U THEN U=Y」である。また，最大値更新処理については，yがvよりも大きい場合にv＝yの処理を実行する必要がある。これに対応できるのは，3の「IF Y>V THEN V=Y」である。

よって，**テは1**，**トは3**になる。

前半のシミュレーション時の2次式の計算や，IF文の細かい条件の違いを見ていく点など面倒な点はあるが，それを除けばとくに難しい要素は見あたらない。後半のIF文の論理の反転は，通常のプログラミングでもよく行われる操作であるため，普段からプログラミングに慣れ親しんでいる人にとっては簡単な設問だったと思われる。

数学II・B

数学II・B　第6問（選択問題）（配点20点）

pを3以上の自然数とする。1以上$p-1$以下の各自然数aに対して，数の列

$a_1,\ a_2,\ \cdots,\ a_{p-1}$

を次のように決める。

- a_1はaとする。
- a_{i+1}は$a_i \times a$をpで割った余りとする。ただし，$1 \leqq i \leqq p-2$である。

また，各aに対して$f(a)$を次のように決める。

- $a_i = 1$となるiが$1 \leqq i \leqq p-1$の範囲にあるときは，そのような最小のiを$f(a)$とする。
- $a_i = 1$となるiが$1 \leqq i \leqq p-1$の範囲に無いときには$f(a) = 0$とする。

pの値を入力して$f(1),\ f(2),\ \cdots,\ f(p-1)$を出力させるプログラムを考えたい。

[方針]　数の列$a_1,\ a_2,\ \cdots$を上の規則によって決めていく過程で
［ア］になればそのiを出力してFORループを抜けだす。
1から$p-1$のどのiに対しても［イ］ならば0を出力する。

この方針に従って，次のプログラムを書いた。

```
100 INPUT "P="; P
110 FOR A = 1 TO P-1
120    [ ウ ]
130   FOR I = 1 TO P-1
140     IF [ エ ] THEN PRINT "f(";A;")= ";I:GOTO 180
150     B = A*B-P*INT(A*B/P)
160   NEXT I
170   PRINT "f(";A;")= 0"
180 NEXT A
190 END
```

注意：　INT(X)は，Xを越えない最大の整数を表す関数である。

(1)　上の［ア］から［エ］に適するものを，次の①～⑦のうちから一つずつ選べ。

①　$a_i \neq 1$　　②　$a_i = 1$　　③　B = 0　　④　B = 1
⑤　B <> 0　　⑥　B = A　　⑦　A = B

(2)　このプログラムを実行する。表示
P=?
に対して7を入力したとき，はじめの4行は
f(1) = ［オ］
f(2) = ［カ］
f(3) = ［キ］
f(4) = ［ク］
と出力される。

(3)　上のプログラムで140行と150行を入れかえたプログラムを実行させ
P=?
に対して9を入力すると，はじめの4行は
f(1) = ［ケ］
f(2) = ［コ］
f(3) = ［サ］
f(4) = ［シ］
となり，意図した結果とは異なるものが出力される。

平成14年度大学入試センター試験本試験より

問題をぱっと読んだだけでは何がどうなっているのか判らない。このようなややこしい問題は，条件を図や表にしてみたり，簡単な例を考えてみたりして，出題者の意図を探る必要がある。

まず，1以上p－1以下の自然数aに対してそれぞれ数列を決めている。ここでは仮にpを5として考えてみる。すると，1，2，3，4の四つの自然数について数列が決まることになる。数列は1項目がaそのもの，それ以後は

ａをかけてｐで割った余りとなっている。ただし，項は１からｐ－１までなので，この例ではａ１からａ４までとなる。計算していくと，

a	a 1	a 2	a 3	a 4
1	1	1	1	1
2	2	4	3	1
3	3	4	2	1
4	4	1	4	1

という具合になる。さらに，それぞれのａについて，ａｉが１になるものがあればその中で最も小さいｉの値がｆ（ａ）となり，無い場合はｆ（ａ）は０となるようにｆ（ａ）が定義されている。よって，

a	a 1	a 2	a 3	a 4	f（a）
1	1	1	1	1	1
2	2	4	3	1	4
3	3	4	2	1	4
4	4	1	4	1	2

ということになる。

このようにして決まるｆ（ａ）の値を出力するプログラムが問題のプログラムとなっている。

設問１ではプログラムの方針とプログラム中の空欄を埋めることが求められている。まず，方針に着目する。アとイはそれぞれｆ（ａ）を出力する時の条件を示している。アの方はｉを出力する条件であり，イの方は０を出力する条件である。よって，イの方は数列の項に１が全く無い場合に該当することがわかり，同時にアの方は１があった場合についてのものであることがわかる。このことから，アに該当する条件は２，イに該当する条件は１になる。

次にプログラムに着目する。方針内容から，まずは数列を計算する部分があり，数列中に１が出てきたらFORループを抜け出すことになっている。プログラムや選択肢の内容から，この「FORループからの抜け出し」が140行のGOTO文であることがわかる。よって，エは該当数列が１になったことを判断する条件式となる。

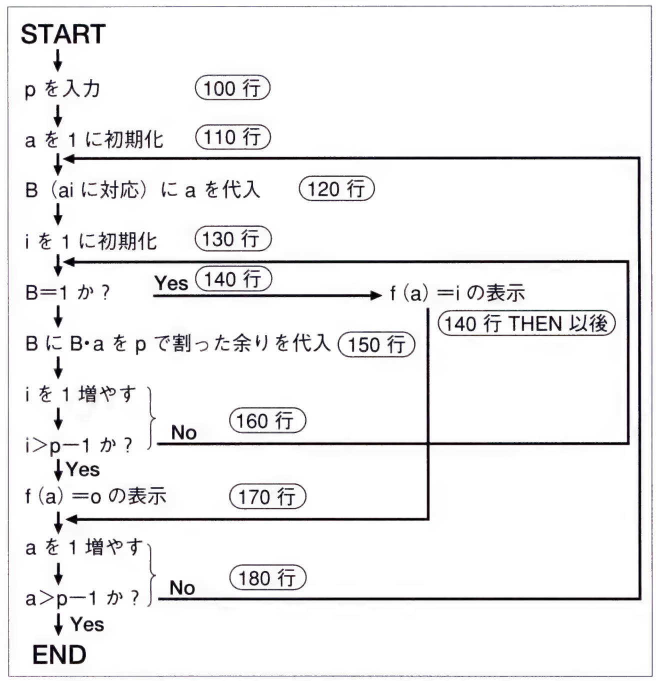
ここで，選択肢を見ると，１との条件判断が行えるのは４の「B=1」だけである。よって，エは４であると予想される。この予想が正しいと仮定すると，変数Ｂは数列が格納されていることになる。これについては，150行の計算が「ＢにＡをかけたものをＰで割った余り」という，数列の項を計算する処理に対応していることから裏付けられ，エは４で正しいことになる。そうなると，ウは数列の最初の数値ａ１をＢに設定していることになり，６の「B=A」となる。

これらのことから，**アは２，イは１，ウは６，エは４**となる。フローチャートは**第3図**のようになる。

設問２では，ｐを７にした時のｆ（１）からｆ（４）が問われている。ここでは，先ほどと同様の表を作成して結果を求めてみる。

a	a 1	a 2	a 3	a 4	a 5	a 6	f（a）
1	1						1
2	2	4	1				3
3	3	2	6	4	5	1	6
4	4	2	1				3

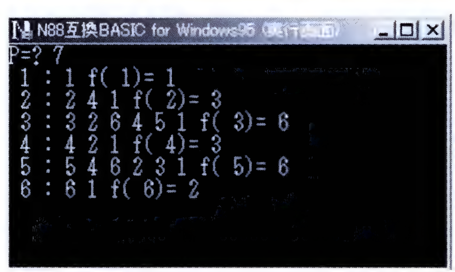
この表から，**オは１，カは３，キは６，クは３**となる。設問３の内容から，設問２の段階ではプログラムが問題に正しく対応していることがわかるため，問題から直接計算したこの表とプログラムの動作は対応していることが確認でき，表の結果とプログラムの出力結果は一致する。問題のプログラムの表示を拡張した**リスト３**を実行した結果は**写真３**のようになり，この表と対応していることがわかる。



≪第3図≫ 数学Ⅱ・Ｂの問題のフローチャート

≪リスト3≫

```
100 INPUT "P=";P
110 FOR A=1 TO P-1
115   PRINT A;" :";
120   B=A
130   FOR I=1 TO P-1
135     PRINT B;
140     IF B=1 THEN PRINT " f(";A;")=";I:GOTO 180
150     B=A*B-P*INT(A*B/P)
160   NEXT I
170   PRINT " f(";A;")= 0"
180 NEXT A
190 END
```



◀**《写真3》 リスト3を実行した結果**
問題のプログラムに数列表示機能を追加したリスト３の実行結果

設問３では，140行と150行を入れ替えたプログラムについて，ｐ＝９とした場合の出力が問われている。プログラムの内容を変更したため，表による結果とプログラムの出力結果は一致しないと考えられる。よって，入れ替えたプログラムの動作に着目する。

140行と150行を入れ替えるということは，数列の数が１かどうかの判断の前に，「ＢにＡをかけたものをＰで割った余り」を計算してしまうことになる。この計算は数列の次の数値を求める計算であるため，ａ１についての評価は行われないことになり，以後の数列ａｉについても本来の値よりもｉの値が１つ大きくなったものに対応するようになる。入れ替えたプログラムでのａ１は，元のプログラムのａ２に対応するわけである。以後の処理は同じであるため，元の数列をａｉ，入れ替えたプログラムでの数列をｃｉとすると，ｃｉ＝ａ（ｉ＋１）という関係になる。

このことを踏まえて，まず正常な表を作成すると，

a	a1	a2	a3	a4	a5	a6	a7	a8	f(a)
1	1								1
2	2	4	8	7	5	1			6
3	3	0	0	0	0	0	0	0	0
4	4	7	1						3

となる。ａ＝３の場合は，余りが０になってしまうためにｆ（３）＝０となる。

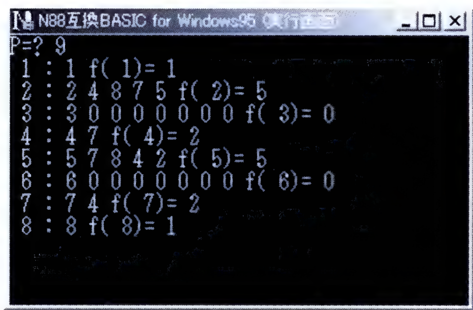
この表から，入れ替えたプログラムでの数列を求めると，

c	c1	c2	c3	c4	c5	c6	c7	c8	f(c)
1	1								1
2	4	8	7	5	1				5
3	0	0	0	0	0	0	0	0	0
4	7	1							2

となる。よって，**ケは１**，**コは５**，**サは０**，**シは２**となる。問題のプログラムの140行と150行を入れ替え，表示を拡張した**リスト４**を実行した結果は**写真４**のようになり，この表と対応していることがわかる。

≪リスト4≫

```
100 INPUT "P=";P
110 FOR A=1 TO P-1
115  PRINT A;" :";
120  B=A
130  FOR I=1 TO P-1
135   PRINT B;
140   B=A*B-P*INT(A*B/P)
150   IF B=1 THEN PRINT " f(";A;")=";I:GOTO 180
160  NEXT I
170  PRINT " f(";A;")= 0"
180 NEXT A
190 END
```



▲≪写真4≫　リスト4を実行した結果
リスト3の140行と150行を入れ替えたリスト4の実行結果。リスト3の表と桁が一つずれていることがわかる

やや複雑な数列の指定から，具体的な数列とｆ（ａ）のイメージをつかめるかどうかがポイントとなる。イメージがつかめていれば，設問２についてはプログラムのシミュレーションをほとんど行うことなく表を作成するだけで解答できてしまう。設問３では処理の順番の変更が与える影響がわかれば，設問２と同様の表を作成するだけの問題になる。FOR-NEXTの２重ループがあったり，そこからの強制抜け出しがあったりと，プログラム的にやや複雑にしてはあるものの，設問の影響で，数列に絡んだプログラムの問題というよりも，プログラムに絡んだ数列の問題という雰囲気になっている。

今年の問題の特徴

プログラムによって制御されたコンピュータは，大容量データの処理や繰り返し計算などを行う場合に，その本質的な性能を発揮する。しかし，センター試験では実際のコンピュータを使えないことから，それらの特徴に絡んだ問題が出題しづらく，出題可能な問題のバリエーションが限られている。その中で，新しいタイプの問題が出題されることもあったが，今回の問題についてはおおよそオーソドックスなタイプになっており，とくにつまづく点は無かったと思われる。

数学Ｉ・Ａについては，シミュレーションに要する時間が長めになることや，条件判断の等号の有無に細心の注意を注ぐ必要があることなど，解答するのに面倒な点が多くなっている。プログラムの技術面ではとくに難しさは感じられないが，試験問題的にはやや難しいと言えるだろう。

数学II・Ｂについては，数列作成プロセスが比較的難しいため，数列の把握がしづらくなっている。ただし，設問自体には難しいものや時間のかかるものは無く，比較的簡単になっている。数列把握ができるかできないかで大きく得点が変わると思われる。

パソコンの心臓部　マイクロプロセッサ誕生30周年企画

CPUの歴史と未来をさぐる！

1979年登場の8ビットCPU「6809」最強伝説編

文・絵／くりひろし

68系というと何を思い出すかな？

つぐ美：うぃっーす遅ようございますぅ。

Dr.D：むむ，つぐ美君にしては影くんより出勤が遅いとは…あ，今日は秋葉原にマーケットリサーチで寄ってから来るって言ってたっけか。ん，なんだ，何買い込んできたんだ？

つぐ美：ふふふっコレ，いいでしょPDA！秋葉原いったら年度末セールがスゴクってさぁ～つい買っちゃったのよぉ。

Dr.D：あ～，買ったんだぁ。さすがはモバイラーつぐ美ちゃんじゃのう…って，Palm系だなそれは？

つぐ美：いけなかったかしら。そりゃ，最近はPocket PC系がノシてきてるのは知っているけど，Palm系の簡便さは捨てがたいのよねぇ…。

Dr.D：いやいや，わしが言いたかったのは，そんなことではなくてじゃな。それって，プロセッサに何入っているか知っとるかね？

つぐ美：さぁ？

影：ドラゴンボールでしょ。

つぐ美：マンガじゃないんだから，いい加減なコト言わないで！

影：ホントだってば…DragonBall VZってゆーモバイル用のプロセッサが入ってるんだってば。

Dr.D：うむ，そのとおり！しかも，メーカーはあのモトローラ社じゃ。

つぐ美：モトローラっつーと，PowerPCとか，68000の？

影：68000も何も，DragonBallって68000系のコア使ってるそうだよ。

つぐ美：それは知らなかったわぁ。

Dr.D：68ナントカっていうと，同じモトローラ社が同時期に作った伝説の8ビット・プロセッサを思い出すのう。

つぐ美：ひょっとして…6809と言いたいのかな？な，なんか最近ゴーインですね，Dr.D？

Dr.D：うむ，ベーマガ編集部のカルロス・ゴーインと呼んでくれたまえ。

前回紹介したZ80は，源流をたどると世界初のマイクロ・プロセッサ4004に行き当たります。もともと高機能なプログラム電卓を開発するために設計された4004とその子孫は，豊富なレジスタセットを持つZ80に代表されるように，良くも悪くも電卓的なつくりになっていました。

ところで，当時コンピュータといえば，電卓と対照的にずっと大型の汎用コンピュータやミニコンピュータが存在していました。インテル社が電卓用LSIから発展させたのに対して，ミニコンなどを小型化する設計思想でプロセッサを開発したメーカーがありました。それが，モトローラ社だったのです。

モトローラ社はもともと通信機器メーカーで(今でもそうですが)，通信機器向けにカスタムのLSIを開発していましたが，これらを汎用的に制御できるカスタムLSIの必要性とインテル社のマイクロ・プロセッサ開発に影響を受ける形で，8008に遅れること2年後の1974年，世界第二の8ビット・プロセッサ，MC6800が開発されたのです。

その設計には捨てがたいものがあった「6800」

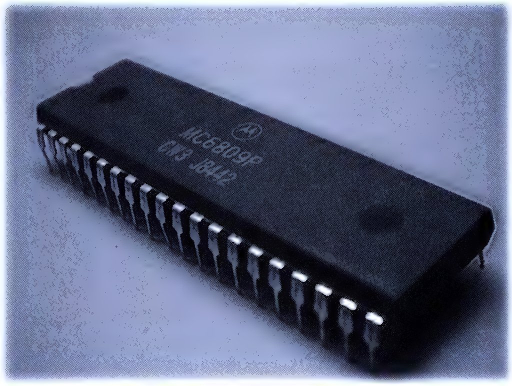
6800はZ80よりも先に開発された割には，8080の持っていた弱点である複数電源を5V単一電源にしたり，相対アドレッシングが可能だったりと，いくつかの改良点が見られますが，その一方でレジスタが少ないために処理が面倒だったり，16ビットデータの演算が苦手などといった欠点も多々あり，TK-80とならんで有名な初期マイコンである富士通LKIT-8な



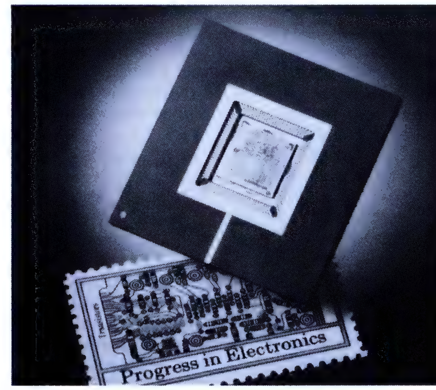
ど，いくつかの機種に採用されましたが，それほど普及しませんでした。

1977年には6800と完全互換で，RAM内蔵によりボードマイコンへの搭載に向いた，6802(インテルで言えば8085相当）が発表されています。

6800は今ひとつの成績ではありましたが，ミニコンを凝縮したような設計には捨てがたい物があります。モトローラ社はボードマイコン，組み込み用のMC6802や周辺チップをいくつか開発し，遂に1979年，Z80に遅れること3年，後に究極の8ビット・プロセッサと呼ばれるMC6809を発表します。



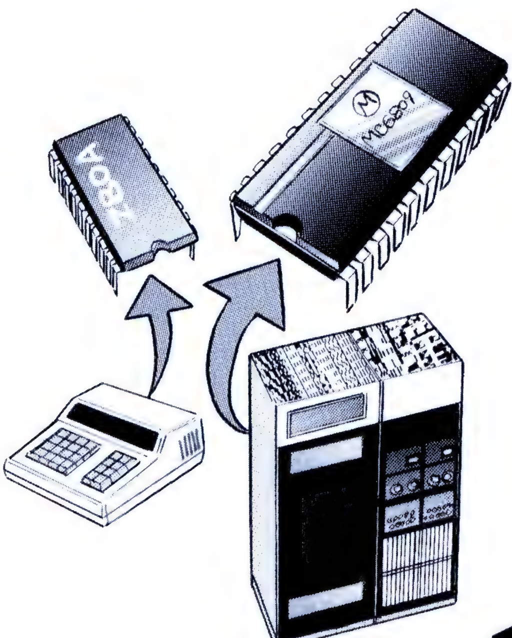
▲68系CPUで1979年登場の「6809」。究極の8ビットCPUと呼ばれた



▲68系CPUの32ビット化がこの「68000」。後にシャープから発売されたパソコン「X68000」がブームに

6800系搭載パソコン一覧

年	メーカー	機種	搭載プロセッサ／特徴
1975	Motorola	MEK6800D	MC6800（トレーニング用）
1976	富士通	LKIT-8	MC6800
1977	日立	H68／TR	HD6800（MC6800互換）
	富士通	NEW LKIT-8	MB8861（MC6800上位互換）
1978	日立	ベーシックマスター	HD6802（MC6802互換）



「6809」で花が咲く！ 究極の8ビットCPUのはじまり

Dr.D：つぐ美君はMPUという言葉を知っておるかな？

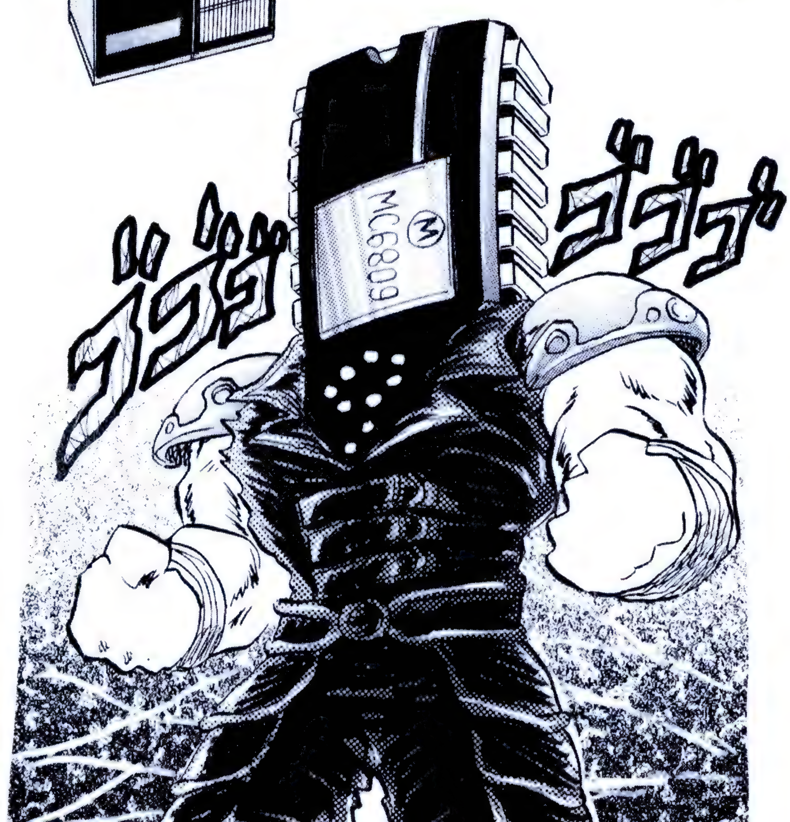
つぐ美:んー，なんか聞いたことあるような…。

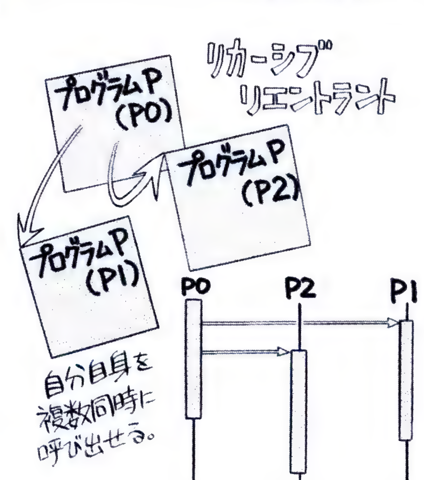
Dr.D：MPUとはMicro Processing Unitの略称でな，モトローラ社ではプロセッサのことをMPUと呼んでいたんじゃ。意味的にはCPUと変わらないのだが，今ではあまり使われなくなった言葉じゃのう。

つぐ美:そういえば，6809と同時期に，モトローラ社の16ビット・プロセッサとして，あの68000が発表されているのよね。そういう意味では，Z80も息の長いプロセッサだったけど，68000も長いのねぇ。

Dr.D：残念ながら，6809の子孫は途絶えておるようじゃがな。まぁ，6809と68000はイトコのようなもんだから。

6809はミニコン思想が出発点だったので，相対アドレス指定によるプログラム・コードの再利用性やマルチタスク向けの高度な命令が豊富でした。電卓思想が出発点のZ80では，レジスタは豊富なものの，アドレッシングは逆に非常に乏しか





6809の特徴

- 67個と豊富なアドレッシング・モードの搭載
- 16ビット演算命令を追加
- レジスタ強化その1：インデックス・レジスタを二つに
- レジスタ強化その2：ポインタ・レジスタを四つに
- リカーシブリエントラントなどマルチタスクOS向け命令の搭載
- 6800とはアセンブラ・レベルの互換性を持つ（バイナリ非互換）
- 相対ジャンプでプログラムのリロケータブル（再配置性）を実現
- Z80同様の5V単一電源による簡便性確保

ったので，まさに対照的なプロセッサだったといえるでしょう。ただし，6809では6800に対して弱点だったレジスタの数も増やし，68系ならではの長所とZ80の長所をも併せ持つことができたのです。

実際，プログラミングをする場合でも(この場合，アセンブラによるマシン語プログラミングのレベル。BASICではプロセッサではなくパソコン全体の機能による差異が大きい)，6809を使うのとZ80を使うのではずいぶん書きかたがちがい，どっちが書きやすかったは，いまだに論争になるほどです。

しかし，プロセッサ同士の同じ機能で比べた場合，ほとんどの部分で6809はZ80より高速に動作しました。究極の8ビットの名前はダテではなかったのです。

68系プロセッサ搭載機種の数々

つぐ美：6809搭載のパソコンっていうと，思い出されるのはまずFM-7よね！

影：ああ，タモリのCMね。

Dr.D：わしだったらやっぱベーシックマスター・シリーズじゃの。なにしろ，ワシらが本誌はベーシックマガジンじゃからな！読者投稿プログラムも結構あったしのう。

つぐ美：ふーん，つぐ美ちゃんお子サマだったからわかんないや。

影：まーた言ってるよ，このヒトは…。

つぐ美：あ，でも，FM-8のバブルメモリカセットの空きスペースってさ，小物入れとか灰皿に使えたんだよね。

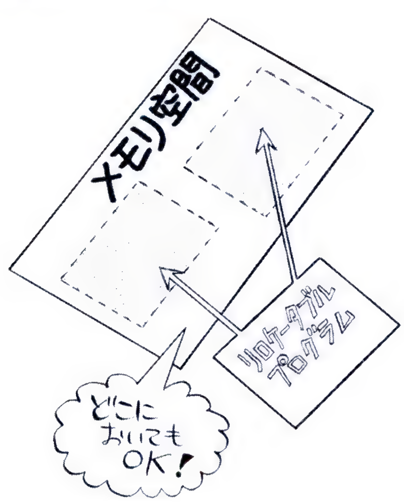
Dr.D：未成年者が喫煙してはいかぁぁんっ！！

Z80程ではないものの，6809を搭載したパソコンは結構ありました。MSXによってZ80が市場を占める前までは，8ビット・パソコン市場はZ80系のNEC・シャープと，6809系の富士通・日立で市場を二分されていました。

とくに6809を印象づかせたのが富士通のFM-7/8です。6809をメインCPUとサブCPUとして二つ搭載することで処理の高速化をはかった，画期的なマシンだったのです。

現在ではマザーボードによってはデュアル・プロセッサ対応のマザーボードもあるし，デュアル使用を前提にしたXeonプロセッサなどがあり，デュアル構成であることを意識せずに利用できますが，当時のデュアル6809では，いかにサブ側の6809とメイン側で処理を分け合うかは，プログラミング・テクニックしだいであり，まさしく腕の見せ所でもあったのです。







▲富士通「FM-7」

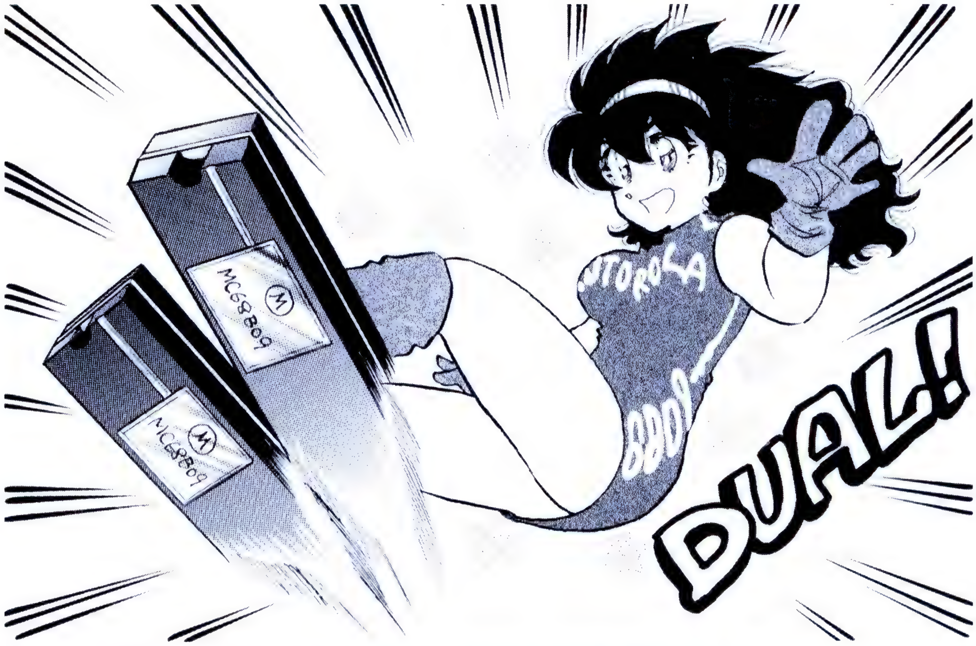
▲日立「ベーシックマスターレベル5」

6809系搭載パソコン一覧

年	メーカー	機種	搭載プロセッサ／特徴
1978	タンディ	Color Computer	MC6809E
1979	日立	ベーシックマスターレベル2	HD6809
1981	日立	ベーシックマスターレベル3	HD6809
	日立	ベーシックマスターJR	HD6809
	富士通	FM-8	MBL68A09，MBL6809の2個搭載
1982	富士通	FM-7	MBL68B09を2個搭載
	松下電器	JR-100	NM1800（MC6802互換）
	日立	ベーシックマスターレベル3マーク2	HD6809
1983	松下電器	JR-200	NM1800（MC6802互換）
	日立	ベーシックマスターレベル5マーク2	HD6809
	富士通	FM-11	MBL38B09Eをコプロセッサとして
1984	日立	S1/10	HD68B09
	富士通	FM-77	MBL68B09E，MBL68B09の2個搭載
1985	日立	S1/40	HD68B09
	富士通	FM-77AV	MBL68B09Eを2個搭載

残念ながら，海外では6809はそれほどメジャーにはなりませんでした。そもそもホビー・パソコンでは大メジャーであったAPPLE-Ⅱはモステクノロジー社の6502を採用していましたし，ビジネス用途には1981年のインテル8088搭載のIBM-PC，さらには同じモトローラ社の68000搭載のMacintoshが出てきてしまい，6809パソコンの出る幕はほとんどなかったのでした。

日本ではビジネス系マシンはPC-9801などの8086系の16ビット，ホビー向けは8ビットと，プロセッサ性能の一種のシバリができていたために，8ビット・プロセッサとしてはZ80以上の性能を持つ6809は，広まる余地があったというわけです。




▲松下電器「JR-100」


▲日立「S1」


▲富士通「FM-8」


▲富士通「FM-77」



もちろん，アーケードゲーム機でも！

Dr.D：さて，ご多分にもれず，6809もゲーム・センターのマシンにも結構使われておったのじゃぞ。それも，案外メジャーどころ！

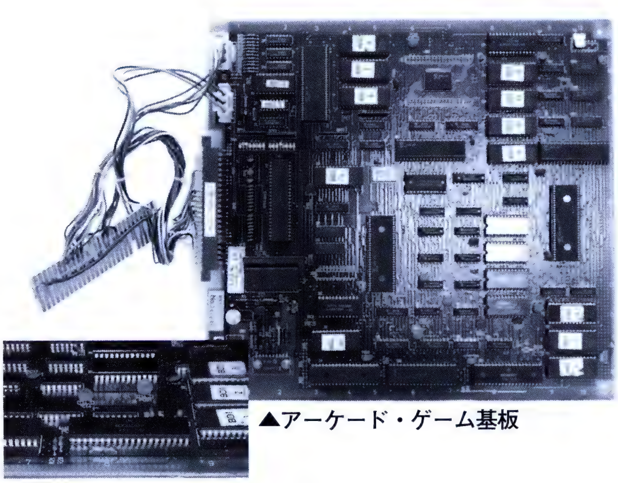
影：ふむふむ，たしかに懐かしいところですね。当時のゲーム・センターで，はじめて見たときに，これは！と思ったのって6809搭載だったんだよね。

Dr.D：記憶がおぼろげなので，ちょっとちがっておるかも。

影：確か，ナムコのSystem 1基板って6809載せてたんじゃなかったかな？あー，ALL ABOUT NAMCOとか思い出しちゃいますねー。源平スキだったなー。笑ってよりとも！

Dr.D：またタモリさんネタかね。ワシはあの黄泉の国の選択が苦手でのう。




▲アーケード・ゲーム基板

6809を採用していたアーケード・ゲーム（一部）

マッピー	6809を2個搭載
リブルラブル	メイン6809，サブ68000
源平討魔伝	6809を2個搭載
ギャプラス	6809を3個搭載

6809バリエーション

6809にも，やはり，動作周波数のちがいや，互換プロセッサ，さらにその改良型などいろいろなバリエーションがあります。Z80ほど多くのセカンド・ソースがあったわけではありませんが，それなりの種類が出ていました。

ちなみに，6809Eの外部クロック供給型は，本来6809には自分で動作クロックを生成することができたのですが，これを排除して，外部のクロックの入力で動作するようにしたものです。これによって，より高速に動作する事が可能になっています。

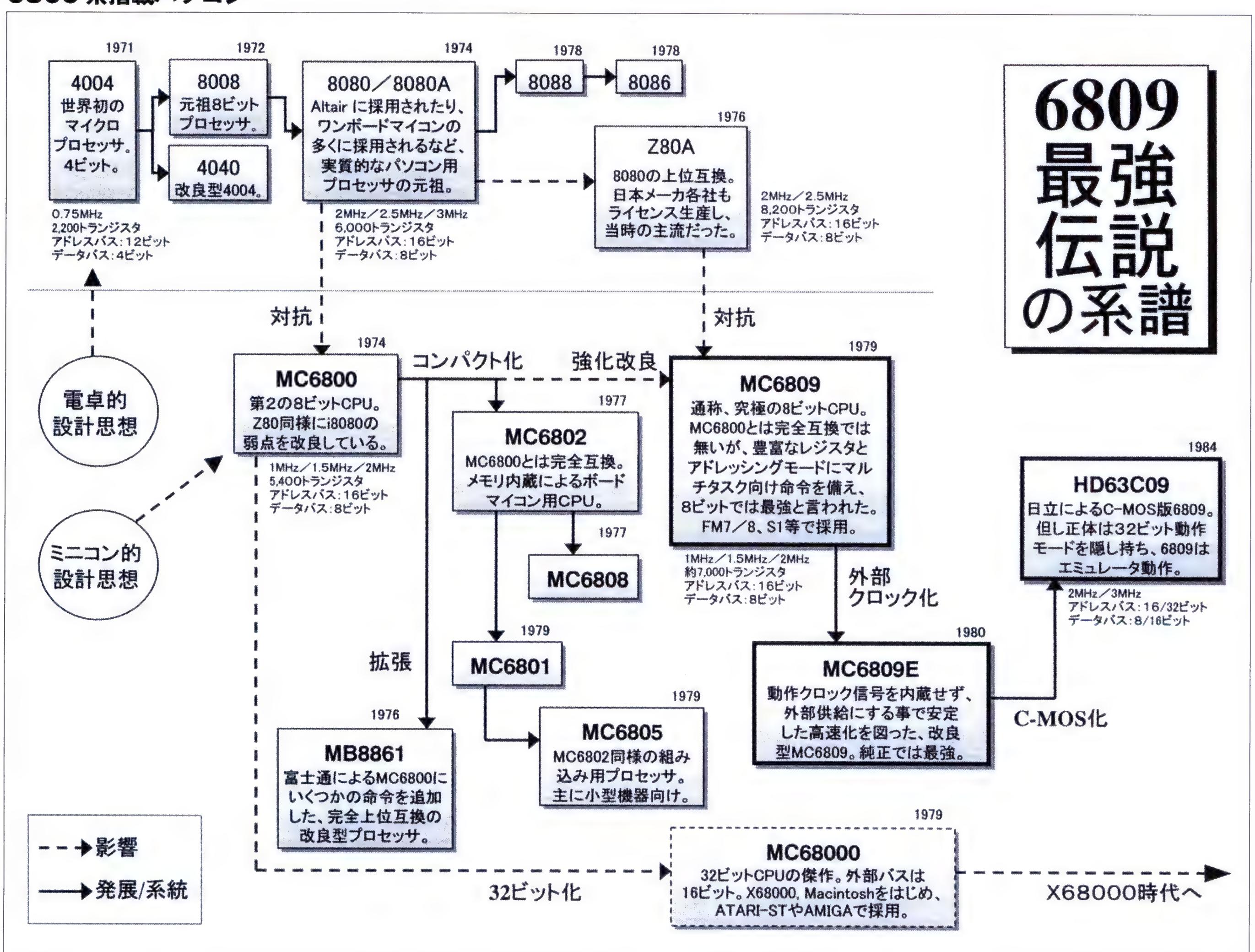
主な国内のセカンド・ソース（ライセンスを受けて正規に互換プロセッサを生産）としては富士通，日立でしたが，い



主な6809のバリエーション／互換プロセッサ

形式名	メーカー	特徴
MC6809	モトローラ	1MHz
MC68A09	モトローラ	1.5MHZ
MC68B09	モトローラ	2MHz
MC6809E	モトローラ	MC6809の外部クロック信号供給型
MC68B09E	モトローラ	MC68B09の外部クロック信号供給型
MBL6809	富士通	MC6809互換，1.5MHz
MBL68A09	富士通	MC68A09互換，1MHz
MBL68B09	富士通	MC68B09互換，4.9/8MHz
MBL68B09E	富士通	MC68B09E互換，2MHz
HD6809	日立	MC6809互換，1MHz
HD68B09E	日立	MC68B09E互換，MC6809互換
HD6309	日立	C-MOS版，32ビットモード内蔵，2MHz
HD63C09	日立	C-MOS版，32ビットモード内蔵，3MHz
HD63C09E	日立	HD63C09の外部クロック信号供給型

6809系搭載パソコン



ずれもFMシリーズにベーシックマスター・シリーズと，自社パソコンに搭載するために生産していた要素が大きいと言えるでしょう。

とくに富士通のFMシリーズは，6809を2個使っていましたから，使用数は自然と2倍！…それはともかくとして，互換プロセッサ・メーカーが独自の機能拡張をしたものはあまり多くありませんでした。しかし，わずかな独自開発の互換プロセッサの中に，6809最強伝説を作り上げる原因の一つでもある，反則に近い性能向上を実現させたプロセッサがあったのですが…。

最強伝説を残した名機の数々

ナゾの男：天が呼ぶ，地が呼ぶ，人が呼ぶ！6809が俺を呼ぶ！

Dr.D：ああっ，あなたは断空我先生！

つぐ美：わぁぁ…って，だいぶ前の読者さんじゃないと知らないかも？

断空我：うるせぇ！ガッツさえありゃ，何だってできるんだ！うおおお！6809最強だっぜい！日立S1サイコー！うぉぉぉぉ！やってやるぜ！

つぐ美：…行っちゃった。

Dr.D：…何しに来たのかな～

最強6809の中でもさらに最強，最強の中の最強が，日立のHD6309です。HD6309は，表向きは省電力にすぐれたC-MOS版かつ3MHz動作版6809だと思われていますが，その実体は16ビット・プロセッサであり，エミュレーション・モードで高速6809として動作します。

また，32ビット・ネイティブモードなるものも備えており，まさに羊の皮をかぶった狼かと思いきや，狼の皮をかぶった虎だったのです。

このHD6309を採用したのが，1984年に発表された，同じ日立のS1（MB-S1）です。通常の8ビット機の直接メモリアクセス空間は64Kでしたが（それ以上は間接的な処理が必要），S1ではアドレスバスを16＋4＝20持つことでメモリ空間のページマッピングを行い，当時としては驚異の1MBのアドレス空間を直接使えたのです。

他にもマシン語呼び出しが高速にできたなど，多くの特徴をもって，S1は究極の8ビット機とも呼ばれたものでした。

S1より少し遅れて，やはり最後の6809搭載パソコンとなる，現在のAVパソコンの元祖といえる機種，富士通のFM-77AV20/40が登場しています。こちらも，当時としては驚異の26万色同時表示を実現していました。

いずれも結局，6809搭載パソコンとしては最後の機種になりましたが，同時期のパソコンの中でも下手な16ビット・パソコンをもしのぐ性能と特徴をそなえていたこと，それらがいずれも「Z80でなく」6809を搭載していたために，6809の「最強の8ビット・プロセッサ」という呼び名は伝説化していくことになります。

忘れちゃいけないOS-9

Dr.D：6809というと，もう一つ，忘れられない特徴がある！

つぐ美：な，なにかしら？

Dr.D：OS-9だ！8ビット・パソコン向けの唯一のマルチタスクOS！

つぐ美：ああ，そんなものがありましたっけ？

マルチタスクとは，基本的に同時に複数のプログラムを実行させることです。初期のWindowsなどでは疑似マルチタスクといって，複数のプログラムを起動させても実行しているのはどれか一つで，他は休止している状態です。

すなわちプロセッサに対する負荷は非常に高く，8ビット・プロセッサではマルチタスクは実現不可能といわれてきました。当時使用できたOSであるDISK-BASICはもちろん，CP-MもMS-DOSも，すべてシングルタスクのOSです。

これに対し，1978年に発表されたOS-9は，6809を対象に，8ビット・プロセッサで，はじめてマルチタスクを実現したのでした。これはもちろん，6809がミニコンを源流とするの当初の設計によるところが大きく，OS-9でマルチタスク・プログラミングをやりたいばかりに，6809搭載のパソコンを買った強者もいたほどでした。

Dr.D：…ってわけで，じつはワシも当時，OS-9使ってみたさに，S1を買ったクチなんじゃよ。懐かしいのぅ。

影：やっぱり…FM7系はゲームなどのソフトもいっぱいありましたけど，それに比べるとS1って，ほとんどなかったですもんね。

Dr.D：だから，自分で作るんじゃよ。なんでもな！そうやって，当時のプログラミング技術は磨かれたもんじゃよ。

影：あああ，なんだか想像しにくいなぁ，脱衣麻雀をプログラミングするDr.Dって！グラフィックとかイジイジと…。

Dr.D：だっ誰がそんなもん作るかーい！

つぐ美：ふ，フケツ！影さんって！

影：…俺かよ！



参考サイト		
	● モトローラ社ホームページ	http://www.mot.co.jp
	● インテル社ホームページ	http://www.intel.co.jp

■連載■ PCハードウエアテクニック

多機能＆多用途パソコンを低予算で導入できる注目テクニック!!

話題のLinuxOS搭載 タワーPCを作ろう

昨年秋に登場したWindows XPをはじめマイクロソフト社のWindowsは，世界で最も利用されているパソコン向けOS（オペレーティング・システム）です。ところが，数年前からWindowsとは少々異なる生い立ちを持つOSが，注目を集めつつあります。それがLinux（リナックス）です。サーバなどのビジネスＰＣや高度な処理にも耐えるワークステーションを破格の費用で手に入れることができるLinux環境を市販のパソコン自作キットで実現してみました。

1 Windowsを超える多機能OS「Linux」とは？

「Linux」という言葉を初めて耳にする人も多いのではないでしょうか。OSに興味のある人なら，数年前からLinuxという文字を，パソコン専門誌などで見かけたことがあるかもしれません。WindowsOSが登場する前から，メインフレームなどと呼ばれる大型コンピュータを中心に利用されていた「UNIX」（ユニックス）というOSがあり，現在でも専門分野のコンピュータに数多く採用されています。じつは，このUNIXに流れをくんでいるのがLinuxで，パソコン向けのUNIXであるPC UNIXの仲間です。

Linuxの特徴のひとつが，ほかのOSに比べ，安い導入費用で済むことです。Linuxは，その名称にもあるように，フィンランド出身のLinus Torvalds（リーナス・トーバルズ）さんが中心になり，開発されたOSで，OSプログラムの内容すなわちソースが一般公開されている「オープンソース」のオペレーティング・システムなのです。

開発に世界中のエンジニアが参加したいきさつもあり，使用は無料で，インターネットでのオンライン配布も行われています。

信頼性と豊富な機能がLinuxの魅力!!

さらに，マルチタスクへの完全対応も特徴となっています。マルチタスクとは，1台のパソコンで複数のアプリケーション（処理作業）を同時に行う最新のOSには欠かせない機能です。

Windowsもマルチタスク対応となっていますが，多少のシステム・エラーでもOS全体の動作に影響を及ぼさないLinuxは，UNIXゆずりの信頼性の高さが特徴です。

Linux標準のサーバ機能を利用する場合は，突発的に発生するアプリケーションやシステムのエラーにも，その都度パソコンをリセットせずに済む信頼性の高さは不可欠で，Linuxは「落ちない」OSなのです。

もう一つ特徴は，UNIXと同様にネットワークの利用を前提にしたOSであることです。複数のパソコン間でデータの交換を行うLAN（ローカル・エリア・ネットワーク）はもとより，TCP／IPプロトコルによるインターネット接続も，他のソフトを使うことなく利用できます。

Linuxには，ネットワーク関連のソフトウエアが含まれています。インターネットでのホームページ公開用に数多く採用されているウエブサーバ・ソフト「Appach」（アパッチ）や電子メールサーバ・ソフト「sendmail」などが，Linuxに

●主要OSの種類と特徴

名称	用途	導入費用	扱いやすさ	ソフトの種類
Linux	ビジネス向けサーバ・アプリケーション開発	極めて安価・オンライン版であれば使用無料	ネットワークOSのため利用にはある程度の予備知識が必要	ビジネスソフト・サーバ(組込)・グラフィックソフトなど・無料のオンラインソフト多数
Windows98・Me	家庭用パソコン向けが主流	1万～2万円台で入手可能	現在の標準的な操作方法で扱いは簡単	ゲームからビジネスソフトまでオンラインおよび市販品多数
Windows 2000	ビジネス向け・サーバ版もある	2万円台から購入可能だが，サーバ版は高価	現在の標準的な操作方法で扱いは簡単。サーバ版はネットワーク関連の予備知識が必要	Windows98・Meと互換性があるが，一部のソフトでは動作に支障あり
Windows XP	一般パソコン・ビジネス・サーバなど	2万円台から購入可能だが，サーバ向けは高価	昨年秋登場したばかりの最新OSで，操作方法は従来のWindowsとは多少異なる	Windows98・Meと互換性があるが，完全対応のソフトの普及はこれから

若松通商 ミニタワーPCキット「Splendor L900Kit」<39,000円>

アップグレード・キットやオリジナルPCキットでお馴染みの若松通商の新製品が,「Splendor L900Kit」(39,000円)。マザーボードは，定評のあるASUSTek社のマイクロATX仕様「TUSI-M」。CPUにCeleron 900MHzを採用し，WindowsOSはもとよりLinux向けマシンとしても，満足できる構成だ。統合チップセット「SiS630ET」を搭載したマザーボードは，ビデオ，オーディオ，さらにLAN機能をオンボードで内蔵し，一般的な用途であれば，拡張カードの追加は不要で，コストパフォーマンスの良さに注目。

主な仕様

CPU	インテル社 Socket 370Celeron900MHz(FSB100MHz)
マザーボード	ASUSTek コンピュータ社　MicroATX 「TUSI-M」 ●対応プロセッサ ：FCPGA / FCPGA2 Pentium III (Coppermine / Tualatin) 450MHz ～ 1.2GHz ／ Celeron 533MHz ～ 850MHz ● チップセット：SiS 630ET，統合 3D グラフィック ● FSB ：133/100/66 MHz ● メモリ：DIMM ソケット×2 （最大 1GB PC133/PC100 ECC SDRAM ） ●拡張スロット： PCI × 3 ／ AMR × 1 ●VGA： Integrated SiS 300 Graphics Engine ● IDE ポート ： UltraDMA/100 ポート×2 ●オーディオ ： C-Media 8738 ● LAN：Chipset Integrated LAN ● I/O ポート ：USB × 2 ／シリアル×1 ／パラレル×1 ／ PS/2 キーボード×1 ／ PS/2 マウス×1 ／オーディオ I/O × 1 ／ゲーム・MIDI × 1 ／ VGA × 1 ● BIOS：Award BIOS
メモリ	128MB SDRAM × 1(PC100)
CD-ROM ドライブ	最大 50 倍速 IDE ドライブ
HDD	20GB
FDD	3.5 インチ 2 モード
ケース	ミニタワー型
キーボード	PS/2 接続日本語 106 キー
マウス	PS/2 接続 2 ボタン



※ハンコム社「LinuxOffice1.5」は別売

は標準実装されています。

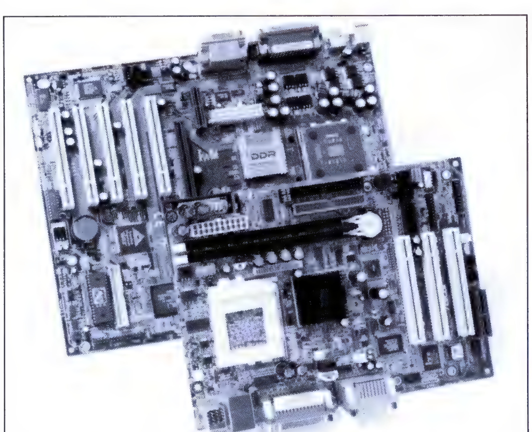
また，Windows PCとのネットワーク接続が可能になるSMBサーバ機能もあります。

使用無料でインターネット·サーバやLANサーバにもなるLinuxは，超多機能OSといってもよいでしょう。

弱点といえば，一般的なビジネスやゲームなどのホビーソフトなどが，Windowsに比べ少なく，Windowsと同様なソフト環境を揃えにくいことでしょう。またWindowsとLinuxではソフトウエアの互換性はなく，Linux用に新規にソフトを揃えなければなりません。

最近では，Linux用のビジネス・ソフトも少しずつ増えています。今回用意した統合ビジネス・ソフト，ハンコム社「Linux Office」もそのひとつです。

Windows用には「Microsoft Office」という表計算やワープロなどをまとめたビジネス·ソフトが有名ですが，これとそっくりなインターフェースと機能を持つのが，Linux Officeです。しかも，1万円台という破格ともいえる低価格で，Linux PCをビジネス用途で使いたい場合は，これだけで十分です。また，フリーウエアの豊富さは，Windows以上の魅力があります。アプリケーションのソースも同時に公開されている場合も多く，プログラミングの知識がある人なら，それをもとに，プログラムの書き換えも可能です。プログラムの作成に欠かせないコンパイラなどの開発環境までもLinuxに含まれています。


▲「Splendor L900Kit」で採用の「TUSI-M」は，コンパクト・サイズのマイクロATX仕様マザーボード。ビデオ，LAN，オーディオ機能をオンボードで装備

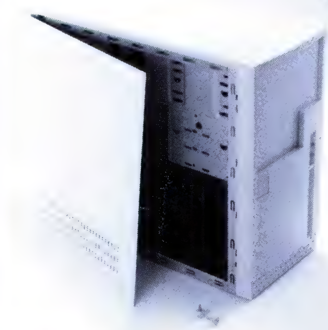
Linux導入に最適なパソコンのハード仕様は？

とくにビジネスやサーバ・ソフトなどで要求される信頼性の高い安定した動作には，Linuxは最適です。とはいえ，パソコンのハードウエアも可能な限り信頼性あるパーツを使用したいものです。最近のように，3D性能を追求したグラフィック・カードや高クロックCPUを搭載したハイエンドPCが人気を集めていますが，高性能を求めるあまりに信頼度に欠けることもあります。

Linux向けのマシンでは，性能の高さ以上に，最新のパーツよりも，実績のあるパーツを揃えることが重要です。ビデオカードや各インターフェース・

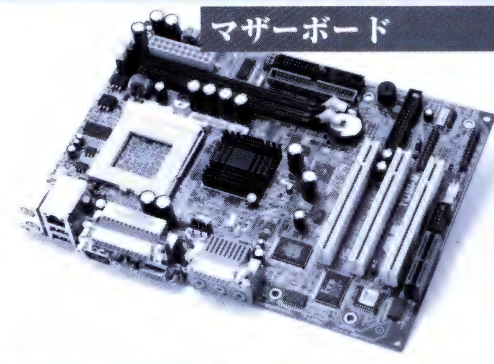
●Linuxマシンの主パーツ選択基準

PCケース



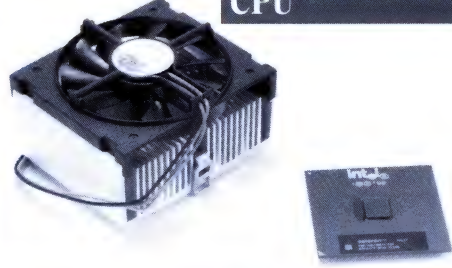
Linuxは，SCSIやIEEE1394などのインターフェースにも対応している。組み立て後のカード拡張などを考慮すれば，ケース内部へのアクセスのしやすさも，PCケース選択の重要ポイント。ケーブル配線のしやすさも含め，ミニタワー・タイプ以上のケースがお勧めだ。

マザーボード



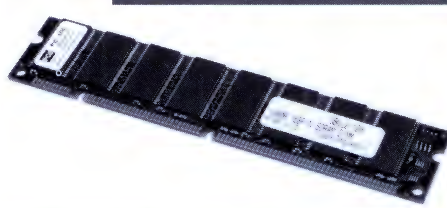
高速性能を求める以上に信頼性の高い実績のあるマザーボードを選びたい。ASUSTek社「TUSI-M」は，Socket370のPentium Ⅲ，Celeron対応マザーだ。

CPU



Windowsよりも軽快に動作するLinuxだが，やはり高クロックのCPUが有利。サーバなどの用途では，1GHzクラスのCPUを選択したい。「Splendor L900Kit」で採用のCeleron 900MHzなら十分だ。

メモリ



Windowsと同様にLinuxも，メモリ容量が多いほど，PC全体の処理速度の向上をはかることが可能。最低で128MB，できれば256MB程度はほしい。

ハードディスク

コンパイラなどのLinuxのプログラム開発環境までインストールすると，OS関連だけで1GB近くの容量になる。そのため，Ultra DMA100対応の高速・大容量のHDDを選びたい。

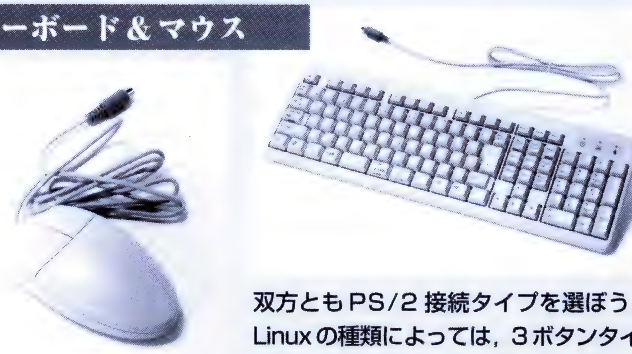


CD-ROMドライブ



LinuxやアプリケーションのインストールにはかせないのがCD-ROMドライブ。できるだけ高速のIDE仕様の製品を選ぼう。

キーボード＆マウス



双方ともPS/2接続タイプを選ぼう。Linuxの種類によっては，3ボタンタイプのマウスが使いやすい場合もある。

カードなどの導入に必要なドライバ・ソフトは，Windows向けよりも開発が遅れがちで，ハードウエアによっては未サポートということも少なからず見受けられます。そのため，最新のものよりも多少旧タイプのパーツを選んだほうが無難です。

今回用意した若松通商から発売のPCキット「Splendor L900 Kit」は，ASUS社の「TUSI-M」（SiS630ETチップセット）マザーやオンボード・ビデオ「SiS 300」など，信頼性で実績を積んだパーツ類で構成されています。CPUにインテル社Celeron 900MHzを採用しLinuxを導入するには，十分すぎるパフォーマンスを備えています。

各パーツを個別に購入し，オリジナルPCを自作する手もありあますが，この「Splendor L900 Kit」が，40,000円足らずで購入できること考えると，低予算でLinuxを導入したい人には，ベストな選択といえるでしょう。

2 PCの組み立てとLinuxOSのインストール

Windowsと Linux向けで，製作方法の違いはありません。

あえて挙げれば，各パーツの組み立てや接続をさらに確実に行うことと，Linuxでサポートしていない，または不明な拡張ボードなどは，極力使わないことでしょう。Linuxに限らず一般にいえることですが，空きスロットに余裕があるからといって，目的もなく拡張ボードを増やすと，突然動作が不安定になることも少なくないようです。

Linuxの信頼性を第一に考えるなら，必要最低限のハードウエア構成にすることがコツです。

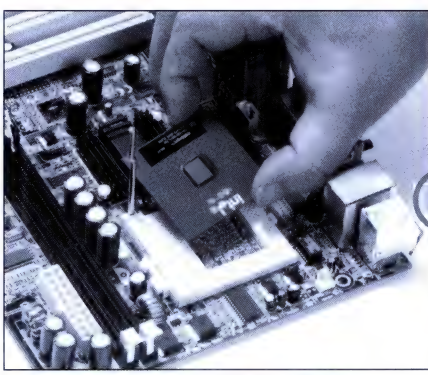
不要なトラブルを回避するパソコン組み立てのコツ

今回，Linuxを導入するPCキット「Splendor L900Kit」は，マザーボードに必要十分なインターフェイスを搭載し，PCIスロットへのカード増設を行わなくて済むため，Linux導入マシンとしては最適といえるでしょう。

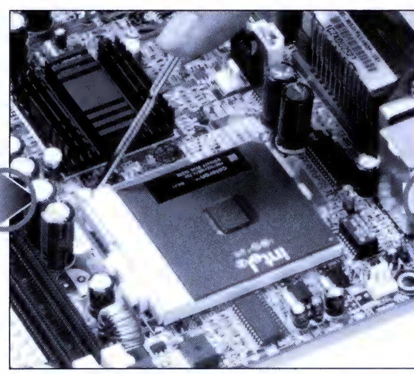
組み立て手順は，マザーボードへのCPU，冷却ファン，メモリの取り付けから始まり，PCケースへのマザー組み込み，さらに各ドライブの装着と配線という流れで行います。このなかで，CPU冷却ファンとメモリの装着のふたつは，多くのトラブルの要因にもなる注意すべきポイントです。高クロックになるほどCPUの発熱量は急激に増加します。完成後に「動作はするものの，時間がたつにつれて不安定になる」といった症状は，冷却ファンとCPUの密着が不十分で，廃熱が上手く行われていない場合に多く見られるようです。

本PCキットのマザーボードのCPUソケットはSocket370タイプで，冷却ファンの取り付けに多少のコツがあります。CPUソケット両サイドの爪にファンのロック金具を引っかけ，固定するわけで

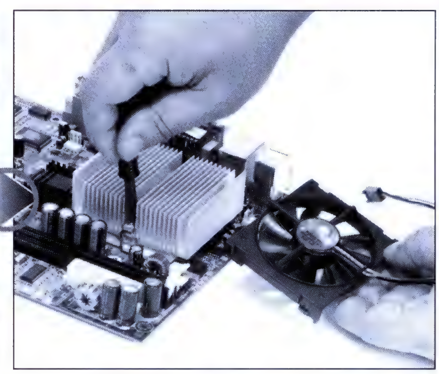
CPUと冷却ファンの取り付け



▲いつものことだが，CPUの装着はソケットとCPUのピン配置を確認しながら行うこと。間違えるとピンが折れる場合もある

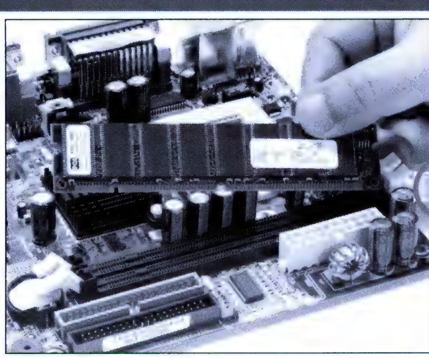


▲CPU装着後は，ソケットのロックレバーを水平方向に倒しCPUを固定する。接触不良にも成りかねないので要注意!!

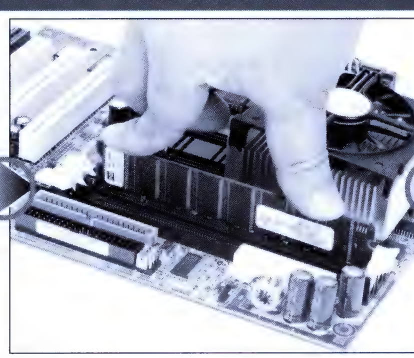


▲クーリングモータ部をいったん取り外し，先にフィンをソケットに装着すれば作業がしやすい

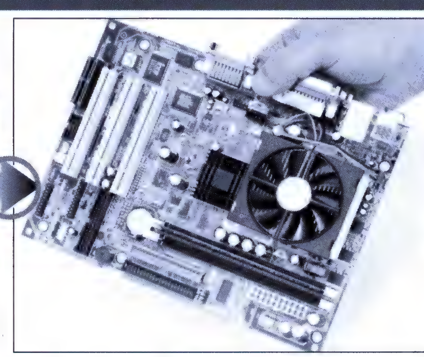
メモリの装着方法



▲メモリのピンとソケットの切り欠き位置を確認し差込み方向を揃える。メモリを扱う場合は，ピンに触れないようにしよう



▲ソケットにメモリを差込んで，左右均等に力が加わるように多少力を入れて押し込むこと



▲CPU，メモリ装着後は，それぞれのパーツが正確な位置に納まっているかを改めて確認しておこう

すが，これがなかなかきつく上手くいかない場合は，冷却フィンからモータ部を取り外したうえで，マイナス・ドライバーなどで上から押すように力を加えれば，意外と簡単に固定できます。モータ部はその後でフィンに取り付ければOKです。取り付け後は，CPU中央のコア部分とファンの接触面が，隙間なく接触していることを確認しておけば安心です。

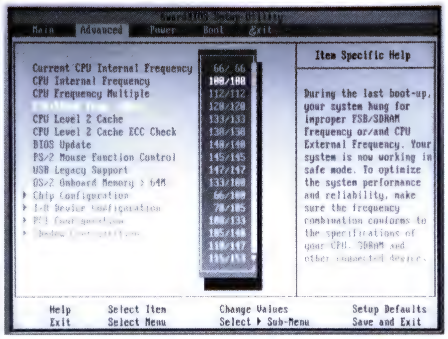
メモリの装着はスロットの奥までしっかりと差込み，両サイドのロックレバーが所定の位置に収まっているかをチェックしておきます。

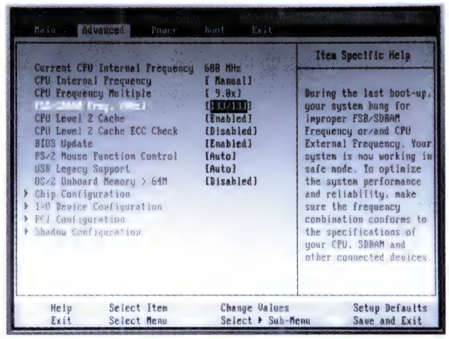
まれなトラブルと思われますが，今回組み立て終了後に電源を入れたところ，パワーランプが点灯するものの，まったく起動できない症状がありました。実は，BIOSのCPUの動作倍率とメモリのFSBが，なぜかCerelon 900MHzとは適合しない数値になっていたことが原因でした。同様なトラブルに見舞われたら，起動時に［DEL］キーを押し，BIOS設定メニューを呼び出し，「CPU倍率設定」と「FSB」設定を疑ってみましょう。

性能には，直接関係はありませんが，動作音の低さもパソコンの優劣を決める重要な要素です。静粛性を求めるために，ハードディスクやCD-ROMドライブをゴム製の振動防止パーツや防音カバーを用いることもあります。手っ取り早く防音効果を高めるには，各ドライブのネジをしっかり締め固定しておくことです。取り外しが簡単にできることから，4か所のネジを3か所だけで済ます人も多いようですが，振動音が大きくなるばかりか，最悪の場合ドライブ自体の寿命すら縮めてしまうこともあります。

所定のネジで確実に固定しておきましょう。

起動しないPCの解決テクニック





組み立て完了後の動作確認で起動しない場合は，BIOSのCPU倍率（右）とFSB（左）の設定が狂っていないかチェックしよう。BIOSメニューは電源投入後に［DEL］キーを押し続ければ表示される

Linuxの入手方法とインストールの手順

Linuxは，インターネットからダウンロードするほかに，パソコン専門誌の付属CDに同梱されている場合も多く，無料または本の購入費用程度で入手できます。Linuxには，「Vine Linux」「Turbo Linux」，「RedHat Linux」といったいくつかの種類があります。これは，ディストリビューション（配布元）メーカー（ディストリビュータともいう）のブランド名で，同梱されたソフトウエアやドライバ・ソフトなどに違いがあるものの，Linuxの基本機能はまったく同じです。また，無料配布版とともに市販パッケージ版もあり，こちらはさらに，有料ソフトウエアやユーザーサポート・サービスが追加されています。

無料配布版の場合，自己責任での利用が原則なので，Linuxを手にするのが初めてという初心者には，市販パッケージ版を利用するのも良い手です。

以前に比べるとLinuxのインストールは，とてもしやすくなっています。ハードウエアの自動認識機能もあり，Linuxがサポートしているハードウエアの範囲内であれば，予備知識がなくてもそれほど難しくありません。

気を付けていただきたいのは，Linuxがネットワーク上での利用を原則としていることから，ネットワーク接続のためのアドレスやユーザー・パスワードの設定が伴うことでしょう。また，不用意な

CD-ROMドライブのPCケースへの取り付け



▲PCケースの種類で，ドライブの設置方法が異なる。そのため，現物あわせで事前に確認しておこう

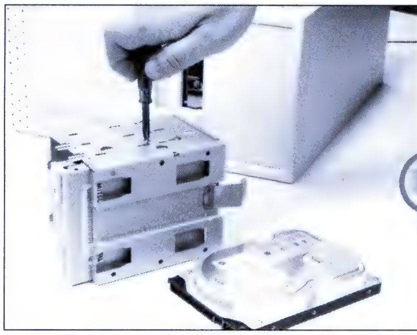


▲取り付け方法を確認したら，設置ベイへ慎重にセットする。きついからといって無理矢理押し込むと，ドライブに悪影響を与える

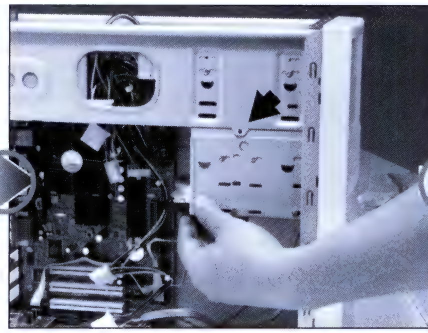


▲動作ノイズを押さえるためにも，手を抜かず所定位置のすべてにネジでしっかり固定すること

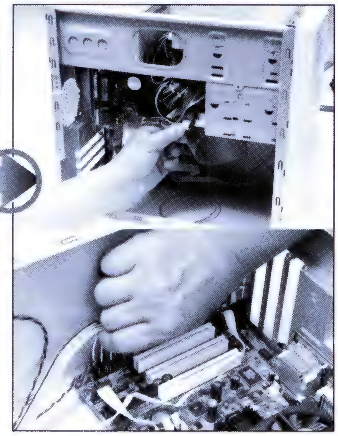
HDD，FDDの装着とケーブル接続



▲今回の「Splendor L900Kit」のように3.5インチ・ベイが取り外せるPCケースは，ドライブの装着が簡単



▲PCケースへHDD，FDD装着済みの3.5インチ・ベイをセットし，専用ネジ（矢印）で必ず固定しておこう。



◀各パーツを所定位置に取り付けたら，ケーブルの接続に取りかかる。こちらも，作業→確認の繰り返しで進めていく

OSの改ざんを防ぐユーザー・アカウント管理（パーミション設定）も，ほかのOSよりも厳重に行われています。そのため，Linux導入前に，ネットワーク設定やアカウントなどについての最低限の知識はマスターしておきたものです。

インストール終了後は，LinuxをGUI（グラフィカル・ユーザー・インターフェース）からコントロールできるデスクトップ環境が起動し，Windowsと同様にマウスによる操作が行えるようになります。このGUIは，X Windows Systemと呼ばれるもので，「KDE」などの複数のタイプがあります。X Windows Systemは，OSインストール後でも変更が可能で，ユーザーの好みに合わせたGUIが選べるのもLInuxの特徴といえるでしょう。

操作や各設定方法は，Windows系OSとは少々異なり，違和感を感じる人もいるかもしれませんが，全体にWindowsよりも軽快に動作するようです。Linux版Netscapeの「Mozilla」（モジラ）でのウエブ表示も，ブロードバンド接続環境であれば，ストレスを感じることなく楽しめました。

さきほどお話しした，統合ビジネス・ソフト「Linux Office」を用意すれば，低予算でインターネット・アクセスからビジネス・ユースまで使える最新OS搭載の多機能PCを手に入れることができるわけです。

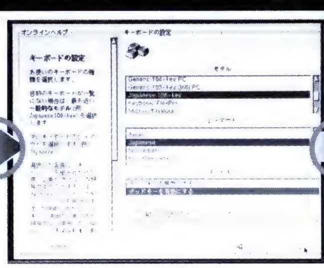


▲Windowsよりも動作が軽快に感じるのは，Linuxの特徴

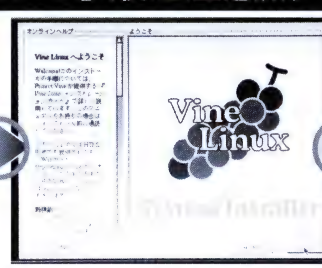
Linuxインストールの手順と注意点



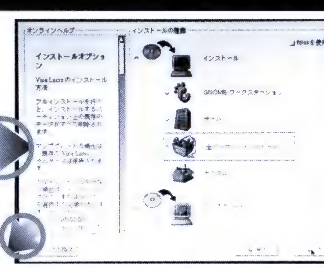
▲インストールCDをセットし，BIOS設定でCD-ROM起動を選ぶと，自動的にLinuxインストーラが起動する



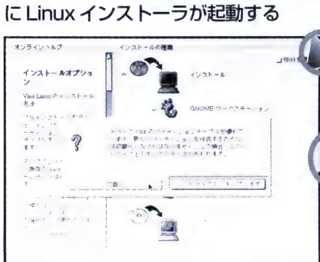
最初の選択メニューは，「キーボードの変更」。通常は「Japanese 106-key」を選べば良い



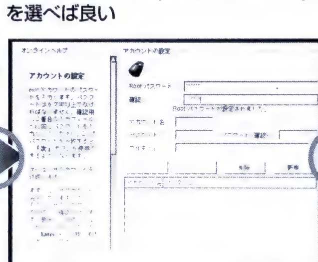
▲キーボードとマウスの選択後，Linuxインストール作業に移る。画面は「Vine Linux」だが内容は他の場合もほぼ同じ



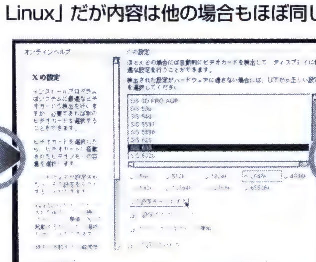
▲用途に合わせ「ワークステーション」，「サーバ」，開発環境も含めた「フルインストール」から選ぶ



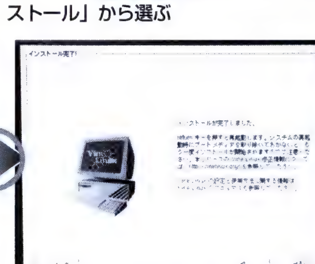
▲Linux用のファイル。システム「ext2」でHDDをフォーマット。パーティション設定も可能



▲OSへのログインに必要なアカウント名とパスワードの設定を行う。忘れてしまうと，Linuxが利用できなくなる



▲ビデオカードのチップ設定はとくに重要。自動認識が可能だが，間違うこともあるので，しっかり確認しておこう



▲フルインストール時は，30〜40分ほど掛かる。この画面が表示されれば，インストール完了だ

PCとAV機器の両方に対応した注目のソニー新型モデル

快適パソコン環境を実現する多機能PCディスプレイ



15インチTFT液晶モデル
SDM-N51AV



17インチCRTモデル
PED-W17M

大抵の場合，パソコン本体よりも末永くつき合うことの多いパソコンのディスプレイ。それだけに製品選びには，目にやさしく，画面クオリティの優れたもの選びたいし，デザインにも気を配りたいものです。ソニーから発売されたばかりの，TFT液晶タイプ「SDM-N51AV」とCRTタイプ「PED-W17M」は，定評のある同社の最新モデルだけに，魅力的な性能，機能，デザインを備えた多機能PCディスプレイです。

パソコンにはPCディスプレイ，番組チェックはTVディスプレイで，というように2台のディスプレイを使い分けている人も多いのではないでしょうか!?

両方のディスプレイを置くには,自分の部屋は狭すぎるという人には,チューナー付のPCディスプレイが，もってこいです。とはいえ，PC専用ディスプレイよりも高価で，しかもTV機能の追加で大型になりがちです。

この点，ソニーから新発売の「SDM-N51AV」と「PED-W17M」は，ディスプレイ部とコントロール·ユニットを分離したセパレート方式を採用することで，ディスプレイはパソコンのそば，コントロール·ユニットはラックに設置というように，省スペースで利用できます。さらに，地上波TVチューナも付いている多機能PCモニタで，リーズナブルな価格なのも魅力です。

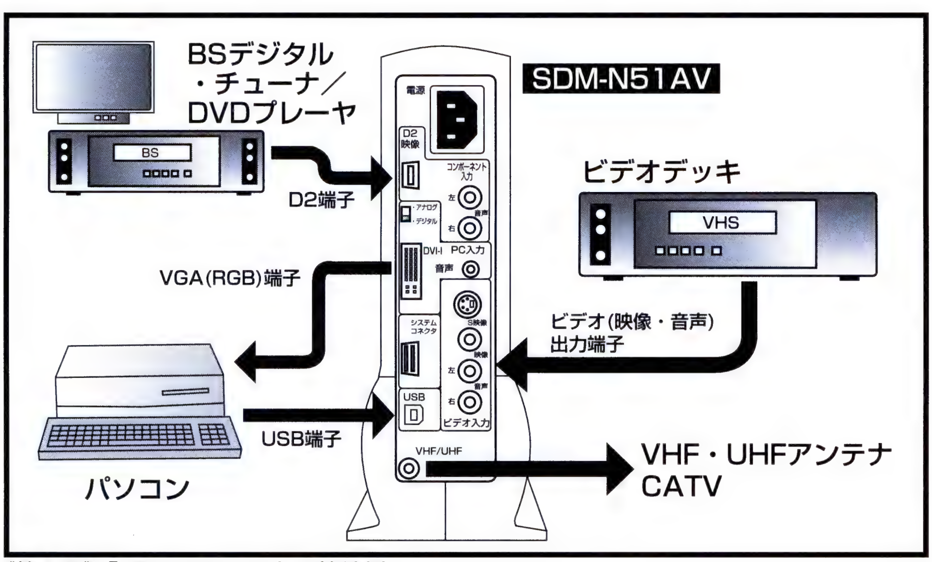
また,地上波放送とは比較にならないほどの高画質な画面で番組が楽しめるBSデジタル放送用のディスプレイとしても使えるD2入力端子など，多彩なメディアに対応できる点も見逃せません。

多彩な使い方ができる液晶モデル「SDM-N51AV」

液晶パネルとスタンドをアームでつなぐユニークな機構を持つ「SDM-N51AV」は，15インチのTFT液晶ディスプレイ·モデルです。ユーザーの体型に合わせ,画面の角度調整が行えるディスプレイは珍しくありませんが,「SDM-N51AV」は，角度調整だけでなく液晶パネルの高さ調整も，アーム＆チルト機構を採用することで可能となっています。

たとえば,パネル角度を垂直方向のままで目線まで高くするなど,セッティングの自由度も広く,長時間の使用にも耐える疲れの少ない角度調整も簡単です。これもエルゴノミック（人間工学）とデザインの両方を配慮した結果といえるでしょう。

16インチ相当の表示サイズを持つ液晶パネルに，高輝度タイプを使うことで，CRTディスプレイにも劣らない，シャープで高画質な表示を実現しています。解像度は最大1024×768で，ビジネ

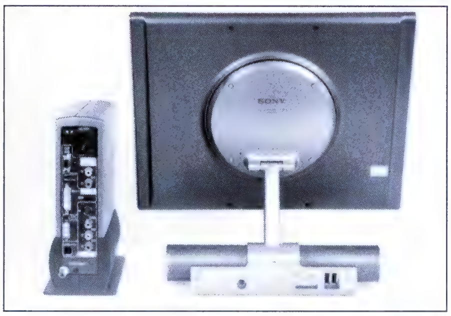


《第1図》「SDM-N51AV」の接続例

多機能PCディスプレイで快適パソコン環境を実現!!

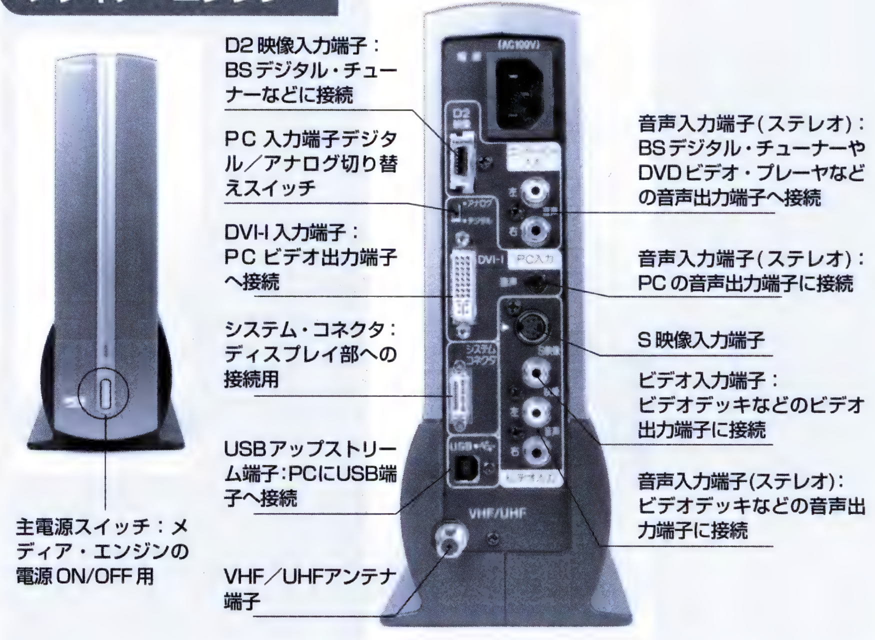
SDM-N51AV





▲省スペースにも役立つ，ディスプレイ部とメディア・エンジン部に分離したセパレート・タイプ

メディア・エンジン

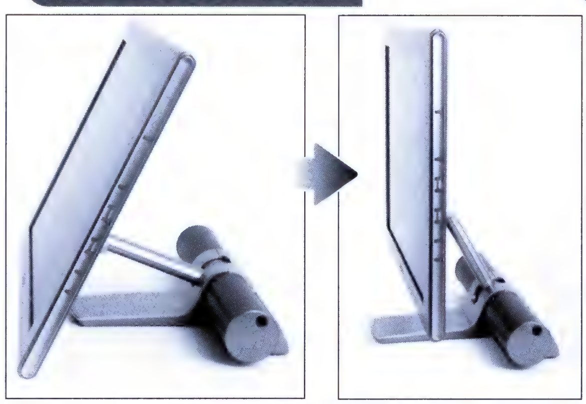


立体音響システム



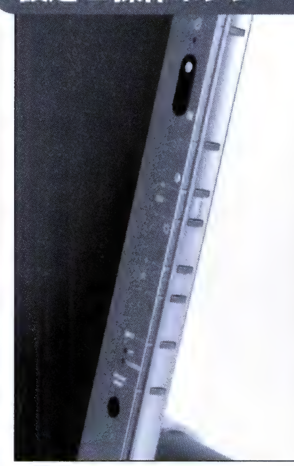
▲スタンド基部はステレオ・スピーカ内蔵。「WOW」システム搭載で，2スピーカーで立体音響を作り出せる

アーム&チルト機構



▲パネルとスタンド基部をアームでつなぎ，パネルの角度や高さの微調整が可能

設定&操作ボタン

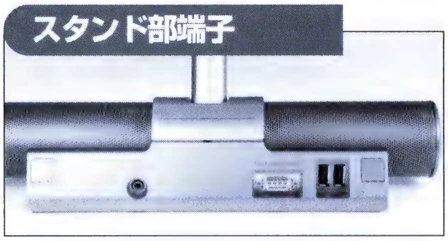


▲液晶パネルの右サイドに，上から電源，メニュー，コントラスト，音量，入力切替，画質モードの各スイッチを配置。画面の詳細設定は画面でメニュー表示を確認しながら，これらのスイッチで行う



◀電源、入力、チャンネル、音量などの切り替えが可能な小型リモコンを付属

スタンド部端子



◀スタンドにヘッドホン，USB×2の端子を装備。マウスなどのケーブル取り回しに便利

スからゲーム・ソフトまで満足できる仕様となっています。また，それ以下の640×480や800×600モードでは，画面サイズに合わせた拡大表示を行うズーム機能が用意されています。

ディスプレイ部と接続するコントロール・ユニットにあたる「メディア・エンジン」は，その名称どおりにパソコン以外にも，さまざまなAVメディアと連携できる豊富な端子を装備しています。

ビデオデッキやDVDビデオプレーヤなどとの連携向けにステレオ音声およびビデオの入力端子のほか，デジタルAV機器向けの高画質な画面表示に対応するD2端子も用意されています。

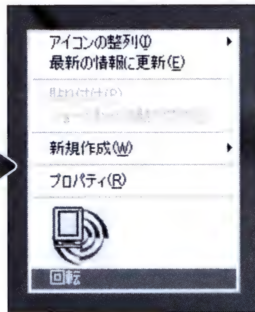
このD2端子は，話題のデジタルBS放送，ハイエンドDVDビデオ・プレーヤなどに採用のビデオ接続規格で，これまでのビデオ入力よりもワンランク上の劣化の少ない高画質ディスプレイが特徴で，コンポーネントビデオ端子とも呼ばれています。ただし，本ディスプレイは，デジタル・ハイビジョン放送規格の表示には対応していません。

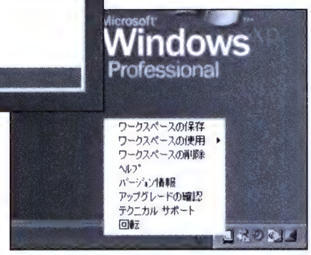
縦長ポートレイト表示になる使い勝手抜群のピボット機能

忘れてはならない魅力の一つが，「SDM-N51AV」のパネル回転機構です。パネルを90°回転することで，768（横）

Web表示やビジネス系アプリに最適!!「ピボット」機能でポートレイト表示対応







▲▼デスクトップ上からカーソルメニューまたはタスクトレイで回転を選ぶ



▲縦長のポートレイト表示は、ホームページや頻繁に縦長レイアウトを扱う表計算、ワープロソフトに抜群の威力を発揮する

×1024(縦)の縦長のポートレイト・ディスプレイに変身します。そのための専用ユーティリティ・ソフトであるピボット・ツールをパソコンにインストールしておけば，Windows OSのデスクトップ上のタスクトレイやカーソル・メニューから「回転」を選ぶだけで，横表示から縦表示へ画面が瞬時に切り替わります。

一つの画面で表示しきれない縦長のホームページ表示，さらに表計算やワープロソフトなどでの文書作成には，このピボット機能は極めて快適で，一度使ったら手放せないほどの優れた機能といってもかまわないでしょう。

数々の使い勝手の良さと，省スペース抜群のスタイリッシュなデザインは，ソニーならではディスプレイに仕上がっています。パソコンと共に新規に液晶ディスプレイを購入する場合や，CRTから液晶への買い換えを予定している人には，「SDM-N51AV」であれば，十分に満足できるはずです。

CRTこだわり派向けの高機能モデル「PED-W17M」

高画質化が進むTFT液晶ディスプレイですが，さらなる解像度や画面サイズを要求する場合は，CRTディスプレイに一日の長があります。

ソニーのCRTディスプレイ・モデル「PED-W17M」は，17インチの平面ブラウン管を採用したAV機能対応モデルです。AV端子の装備は，「SDM-N51AV」と同様ですが，こちらはビデオ・オーディオ端子を2系統用意し，AV機能の強化がはかられています。また解像度は最大1280×1024で，TFT液晶では手が届かない高解像度表示が可能です。

従来から定評のあるトリニトロンCRTは，発色，解像度とも文句のつけどころのないクオリティで，さらにTV放送を4倍密映像で表示するソニー独自の「DRC-MF」搭載や，ハイビジョン信号を入力・表示するD4端子装備など，多機能PCディスプレイではトップクラスの機能，性能を装備したモデルです。

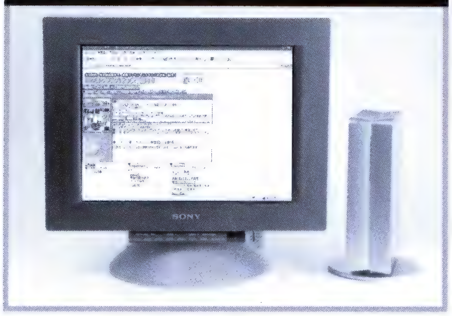
一歩先を行くパソコン・ユーザーには，「PED-W17M」は決して期待を裏切らないはずです。

●「SDM-N51AV」のおもな仕様

液晶パネル・サイズ	15型（CRT17型相当）
方式	アモルファスシリコンTFT方式（a-Si TFT）
最大発色数	約1619万色
コントラスト比	400対1
視野角	140°／120°（水平／垂直）
最大解像度	1024×768（75Hz）
表示画面サイズ	304×228mm（幅×高さ）
水平走査周波数	30～61kHz
垂直走査周波数	48～85Hz
信号入力端子	DVI-I×1（付属ケーブルで15ピンDsubアナログRGB入力対応）,
音声入力	ステレオ・ミニジャック×1
音声出力	内蔵スピーカー 2W×2
ヘッドホン端子	ステレオミニジャック
USB端子	1アップストリーム／2ダウンストリーム
パネル・チルト角	0°～90°（フォトスタンド時：0°～50°）
TVチューナー部	
受信チャンネル	VHF1～12，UHF13～62，CATV C13～C35
アンテナ端子	VHF／UHF 75Ω・F型コネクタ
外部入力端子	ビデオ入力（コンポジット，S映像）×1，コンポーネント入力（D2端子）×1
音声出力端子	内蔵スピーカー 2W×2
電源	AC100V・50／60Hz
消費電力	通常時：39W（テレビ／ビデオ使用時48W） 省電力モード時：10W アクティブオフ時：3W以下
外形寸法	<ディスプレイ部>アーム／画面正立時：364W×343H×190Dmm，アーム最前傾時（50°傾斜時）：364W×236H×256Dmm，<メディアエンジン部>90W×221H×202Dmm
質量	<ディスプレイ部>約3.8kg，<メディアエンジン部>約1.6kg
付属品	電源コード・信号ケーブル（DVI・HD15）・モニターユーティリティ／アプリケーションCD-ROM・新G3／G4専用マックアダプター・専用接続(GVIF)ケーブル・音声接続ケーブル・リモコン・RF同軸ケーブル・USBケーブル

多機能 PC ディスプレイで快適パソコン環境を実現!!

PET-W17M





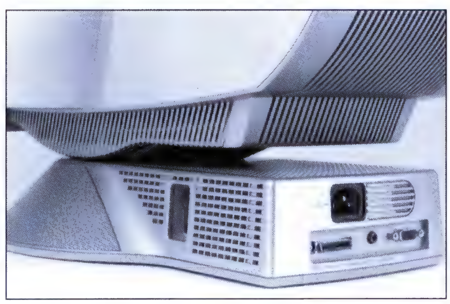
▲多機能でありながら省スペースを確保できるセパレート・タイプ。円錐形のディスプレイ基部は「WOW」対応スピーカーを収納



▲画面の微調整は，画面下のスイッチから行う。「SDM15-N51AV」同様にソースで画質を使い分けるピクチャー・エフェクト機能もある

テレビチューナ・ボックス







▲パソコンのビデオ出力は，ディスプレイ部背面のの入力端子へ接続。▶フラット・ブラウン管の 15 インチ FD トリニトロンの解像度，発色の良さに大注目!!

● 「PED-W17M」 のおもな仕様

CRT サイズ	17 型・FD トリニトロン	外形寸法	<ディスプレイ部>448W × 425H × 434D mm, <テレビチューナーボックス部>, 156W × 268H × 213Dmm
AG ピッチ	0.24-0.25mm		
表面処理	ハイコントラスト AR コーティング		
最大解像度	1280 × 1024 (60Hz)	質量	<ディスプレイ部>約 22kg, <テレビチューナーボックス部>約 3kg
推奨最大解像度	1024 × 768 (85Hz)		
表示画面サイズ	312 × 234mm (4：3)・ 292 × 234mm (5：4) (幅 × 高さ)	付属品	電源コード・信号ケーブル (両端 HD15)・専用接続ケーブル・リモコン・アンテナ接続ケーブル・クリーニングクロス他
走査周波数	水平：30 ～ 70kHz・垂直：48 ～ 120Hz		
信号入力端子	15 ピン 3 列 D-sub	デジタルコントロールシステム	画面調整機能：ブライトネス・コントラスト・水平／垂直位置・水平／垂直サイズ・糸巻歪／糸巻歪バランス・台形歪／台形歪バランス・傾き・色温度調整 (BIAS ／ GAIN)・色温度補正・sRGB・H ／ V コンバージェンス・V コンバージェンス上／下・ランディング調整・モアレキャンセル・デガウス・H ／ V OSD 位置・コントロールロック・ピクチャーエフェクト・fH ／ fV 表示，(プリセットモード：18，ユーザーモード：15)，自動消磁機能：電源 ON 時，自己診断機能：カラーバー
TV チューナー部			
受信チャンネル	VHF1 ～ 12，UHF13 ～ 62，CATV C13 ～ C35		
アンテナ端子	VHF ／ UHF 75 Ω，F 型コネクタ		
外部入力端子	ビデオ入力 (コンポジット，S 映像) × 2，コンポーネント入力 (D4 端子) × 1，AV マルチ入力× 1		
外部出力端子	音声 L・R × 1		
音声出力	内蔵スピーカー 1W × 2		
電源	AC100V・50 ／ 60Hz		
消費電力	通常時　TV モード時：約 135W (最大) PC モード時：約 120W (最大) アクティブオフ時3W以下 (待機電力1W以下)		

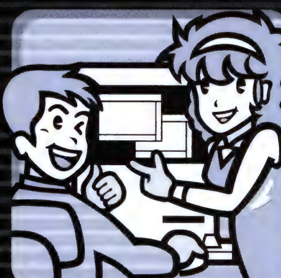
本コーナーは，さまざまな言語に対応したプログラミング学習と創作をテーマに，読者からの投稿作品で成り立っています。これから，プログラミングを勉強しようというみなさんや，プロのプログラマー／クリエイターを目指そうというみなさんにとって役立つ情報が満載です。手近にあるパソコンに入力して，お楽しみください。

それぞれの記事にはプログラムを入力／実行するときの注意点やプログラムの内容，改造方法などが紹介されていますので，よく読んでください。また，"CHECKER FLAG"というコラムがあり，デバッグ博士ことDr.Dがそれぞれのプログラムに対し批評を加えています。口が悪い人なので，かなりキビシイことも言っていますが，自分でプログラムを作ったり，これから投稿しようとしている人にとっては，とても参考になるでしょう。

- Visual Basic
- F-BASIC V6.3
- HSP
- iアプリ
- PC-9801
- FX-890P

BM特選 投稿プログラムコーナー

プログラマー＆クリエイターへの登竜門



プログラム投稿者へのお願い

自分の作ったプログラムを投稿しようという人はつぎのきまりを守って応募してください。

①プログラム（作品）ファイル ②プログラムの説明，住所・氏名・年齢・職業（学校名）・電話番号（連絡のつく時間も）を書いた原稿のテキスト・ファイルをフロッピーディスクなどにセーブし（プリンタで打ち出した原稿も同封のこと），お送りください。

プログラムの説明には，①内容 ②使いかた・遊びかた ③操作方法 ④プログラムのしくみ（変数表やフローチャートも）⑤プログラムの具体的入力方法，注意点 ⑥改造のアドバイス，作ったとき苦労した点 ⑦使用機種名 ⑧言語（バージョン，モードなども）などを書きます。

注1）プログラムは必ずチェックし，ミスを防ぐために同じものを2～3入れてもらえば安心です。

注2）プログラムの説明は，テキスト・ファイルをフロッピーディスクなどにセーブし，念のためプリンタで打ち出したものも同封することが基本ですが，手書き原稿でも受け付けます。その際，できるだけていねいに，マス目のある用紙を使用してください。

注3）ディスクのラベル，封筒の裏面にもプログラム名（ファイル名も），原稿ファイル名，住所，氏名，電話番号を必ず記入してください。

内容はオリジナルのもので，未発表のものに限ります。もし，参考にした記事や本がありましたら，書名，号数，記事名，筆者名，記事のタイトルなども明記しておいてください。

お送りいただいたディスクはお返しできませんので，あらかじめバックアップをとっておいてください。

掲載されたプログラムに対しては，規定の原稿料をお支払いし，掲載誌をお送りいたします。

編集部

★本誌とじ込みの愛読者カードについている応募カードも忘れずに同封してください。

あて先：〒141-8715 東京都品川区東五反田1-11-15　電波新聞社マイコンBASICマガジン「特選プログラム・コーナー」係

掲載プログラムのダウンロード／投稿はこちら

http://www.basicmagazine.net/

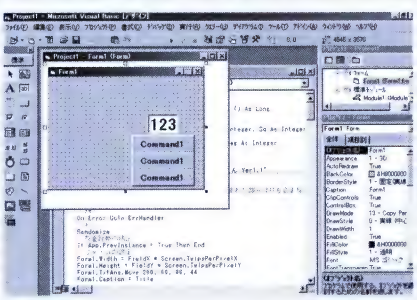
掲載[投稿プログラム]言語

入手方法と入力手順ガイド

Visual Basic

製品紹介

Visual Basic(以下VB)とは，マイクロソフトから発売されている，Windows上で動作するプログラム作成ツールです(開発画面は**写真1**)。現在はVersion6.0が最新版となっています。また，EXE（実行ファイル）作成などの機能を制限した評価版(体験版)も存在します。



▲写真1

入手方法

製品版は，各種類ともパソコンショップなどで購入可能です。学生・学校職員向けに価格を抑えたアカデミックパッケージも用意されています。

また，評価版は本誌で連載していた記事をまとめたムック本『入門Visual Basic 6.0/5.0講座』(税込み2100円)をはじめ，いくつかの書籍に付録として収録されています。

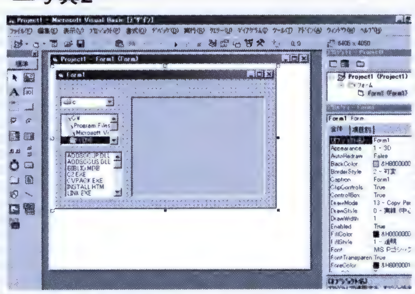
本誌プログラムの基本入力方法

①VBを起動後，標準EXEを選択します(写真2)。

②掲載プログラムのフォームの配置図を



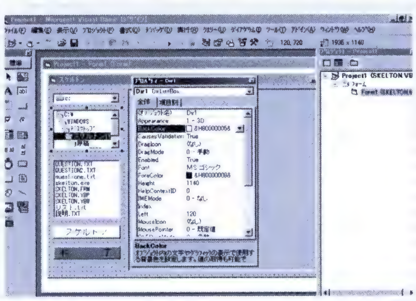
▲写真2



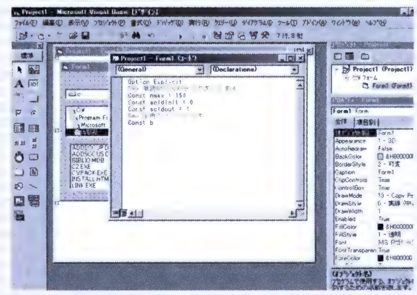
▲写真3

参考にしながら，オブジェクトをフォーム上に配置します(写真3)。

③掲載しているプロパティ表にそって，配置したオブジェクトのプロパティを設定（すでに入っている値を変更）していきます(写真4)。プロパティによっては，別に作成した画像ファイルなどを指定す



▲写真4



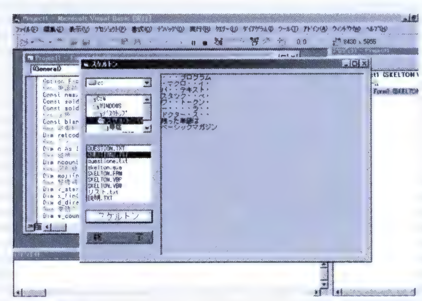
▲写真5

る場合がありますので，その場合は本誌に掲載されている画像写真を参考に，Windows付属の「ペイント」などのソフトを用いて画像を作成します。

④フォームのコードウィンドウを表示し，掲載されているプログラム・リストを入力していきます(写真5)。モジュールのリストがある場合は，同様にモジュールのコードウィンドウからリストを入力していきます。

⑤フォームが複数ある場合は，そのフォームの数だけ，②～④の作業を繰り返します。

⑥すべての入力作業が終了したら，実行ボタンを押しプログラムを実行させます(写真6)。



▲写真6

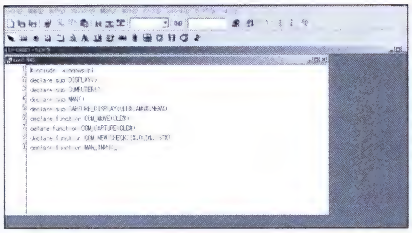
F-BASIC(Windows版)

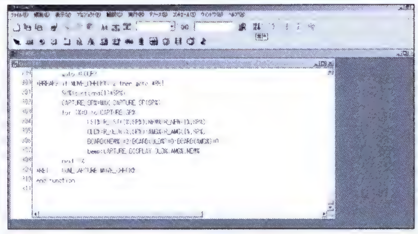
製品紹介

F-BASICとは，富士通ミドルウェアから発売されているWindows用プログラム言語です。現在はVersion6.3が最新版となっています。

入手方法

パソコンショップなどで購入可能です。


▲写真1


▲写真2

本誌プログラムの基本入力方法

①F-BASICを起動後，画面上のエディタから，掲載されているリストを入力します(写真１)。

プログラムによっては，別途画像ファイルを必要とする場合があります。その場合は，誌面に掲載されている画像写真を参考に，Windows付属の「ペイント」などのソフトを用いて画像ファイルを作成します。

②リストを入力後，実行ボタンを押してプログラムを実行させます(写真２)。

HSP

製品紹介

HSPとは，フリーで使用できるWindows用プログラム言語です(開発画面は写真１)。現在はVersion2.55が最新版となっています。


▲写真1

入手方法

HSPは，以下のホームページからダウンロードすることができます(写真２)。

http://www.onionsoft.net/hsp/

また，パソコン雑誌やHSP関連の書籍などの付録として収録されていることもあります。

本誌プログラムの基本入力方法

①HSPを起動後，画面上のエディタから，掲載されているリストを入力します。

プログラムによっては，別途画像ファイルを必要とする場合があります。その場合は，誌面に掲載されている画像写真を参考に，Windows付属の「ペイント」などのソフトを用いて画像ファイルを作成します。


▲写真2

②リストを入力後，実行ボタンを押してプログラムを実行させます。

ｉアプリ

製品紹介

ｉモード対応の携帯電話を使用し実行します。プログラミングはパソコンで行います。

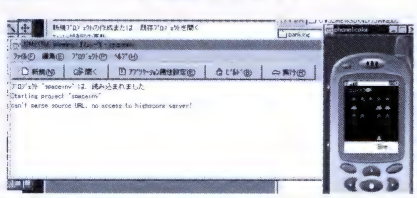
入手方法

携帯電話ショップで購入可能です。

本誌プログラムの基本入力方法

①サンマイクロシステムズのWEBサイト（http://java.sun.com/）からJava2の開発環境「Java 2 SDK Standard Edition 1.3.0」をダウンロードします。インストールすると“C:¥jdk1.3"というディレクトリにJavaコンパイラが展開されます。

② NTTドコモのWEBサイト（http://www.nttdocomo.co.jp/i/java.html）からｉアプリの開発ツール「J2ME Wireless SDK for the DoJa release2.2（以下，DoJa)」をダウンロードします。ダウンロードには名前の登録が必要です。インストールすると“C:¥J2MEWSDK4DOJA"フォルダに主要なファイルが展開されます。


▲写真1

③スタートメニューから「J2ME～DoJa」を選択して，「KToolBar」を起動します(写真1)。

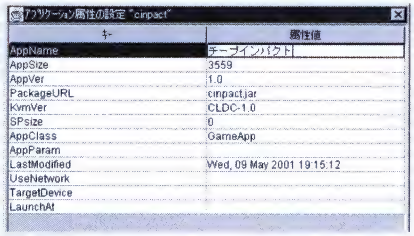
④「新規」ボタンを押して，新規プロジェクト名を入力すると，“C:¥J2MEW SDK4DOJA¥apps"フォルダの下にプロジェクト名のフォルダと，その下に“bin",“res",“src"フォルダが作成されます。

DoJaは，テキストエディタを搭載していないので，別途メモ帳などのテキストエディタソフトでソースファイルを作成します。入力方法は，誌面に掲載されているリストを入力し，指定されたファイル名で保存します。ファイル名を間違えると正しくビルドされないことがあります。また，DoJaの場合，jamファイルは

次のページにつづく

ソフト側で作成できるため，入力する必要はありません。

プログラムによっては，別途画像ファイルを必要とする場合があります。その場合は，誌面に掲載されている画像写真を参考に，Windows付属の「ペイント」などのソフトを用いて画像ファイルを作成します。



▲写真2

⑤“res"フォルダに画像（gif）や音楽ファイル（mld），“src"フォルダにはソースファイル（java）を格納します。
⑥「アプリケーション属性設定」ボタンを押して，キーの値を入力します(写真2)。(“AppSize"と“LastModified"は自動的に更新されます)。
AppName = (アプリケーション名)
AppVer = 1.0
AppClass = (最初に実行するクラス名)
SPsize = (スクラッチパッドのサイズ)
KvmVer = CLDC-1.0
⑦「ビルド」ボタンを押すと，コンパイルとプリベリファイが一括して行われ，jarファイルとjamファイルが“bin"フォルダに生成されます。jarファイルは"res"フォルダの中身とクラスファイルがzip形式で圧縮されたものです。
⑧エディタで“C:¥J2MEWSDK4DOJA¥lib¥devices¥phone1color"フォルダの“phone1color.properties"を以下のように書き換えます（値は120×130ピクセル時の設定です)。書き換えたら「編集ー設定」をメニュー選択して，「デバイス指定」を“phone1color"にします。

```
  :
screen.x=27
screen.y=65
  :
screen.width=120
screen.height=156
  :
icon.up: 44, 138, off
  :
icon.down: 44, 146, off
  :
screenPaintableRegion.width=120
screenPaintableRegion.height=130
  :
softbutton.0=0,138,36,18, softButton
softbutton.1=90,138,36,18, softButton
  :
```

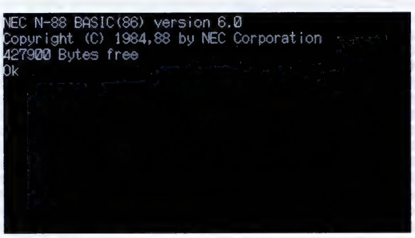

▲写真3

⑨「実行」ボタンを押すと，携帯電話エミュレータ上でｉアプリが実行されます(写真3)。

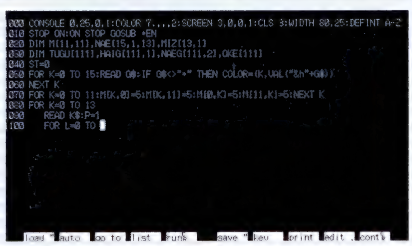
PC-98 N88日本語BASIC(86) DOS/Disk-BASIC

製品紹介

NECから発売されていたPC-9800シリーズ（現在は中古市場で入手可）で動作するプログラム言語です(開発画面は**写真１**)。MS-DOS上で動作するDOS-BASICと専用のフロッピーディスクから起動させるDisk-BASICがあります。両者は基本的な構造は同じですが，ディスクフォーマットの違いや，それぞれに依存する命令が存在します。



▲写真1



▲写真2

入手方法

N88-日本語BASIC(86)(MS-DOS版)(Ver6.2)という製品名で発売されていますが，現在は入手困難となっています。


▲写真3

本誌プログラムの基本入力方法

①N88-BASICを起動後，画面上のエディタから，掲載されているリストを入力します(写真２)。
②入力後，RUNと入力し，リターン・キーを押して実行させます。この操作は[F5]キーでも行えます(写真３)。

FX-890P

製品紹介

カシオ計算機から発売されているポケットコンピュータ(ポケコン)です(本体写真は**写真１**)。本誌のFX-890P用の作品は，主にBASIC言語を使用したプログラムになります。

入手方法

パソコンショップなどで入手可能です。

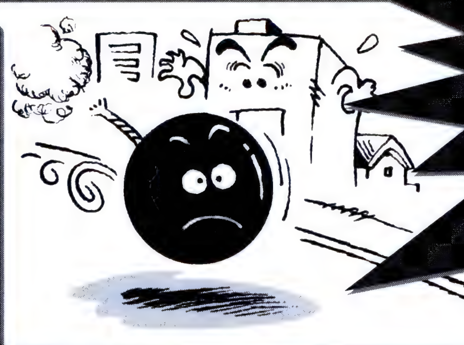
本誌プログラムの基本入力方法

① ポケコンを起動後，MENUキーを押し，表示されたメニューからBASICを選択します([2]キーを押す)。
②BASIC画面になったら，本誌に掲載されているリストを入力します。
③入力後，RUNと入力し，リターン・キーを押して実行させます。



▲写真1

Windows(Visual Basic V6.0)用



懐かしい雰囲気のあるゲーム

チキチキバンバン2

梶原　寿樹（29歳）

入力方法は32ページをご覧ください。

遊び方・ルール

スタート地点(最初に爆弾が表示される所)からゴール地点(スタート地点の対角線上)まで,道をつなげて爆弾を誘導するゲームです。

ノーマルモード：普通にゴールまで誘導します。脱線しても,少しの間は爆発しません。その間に道をつなげてください。

オリジナルモード：BLOCK(壁)にぶつかると反対方向に移動します。 このモードでは，必ず1回以上BLOCK(壁)にあたって方向転換しなければ，ゴールしても爆発します。落とし穴(黒いところ)に落ちると即，爆発します。

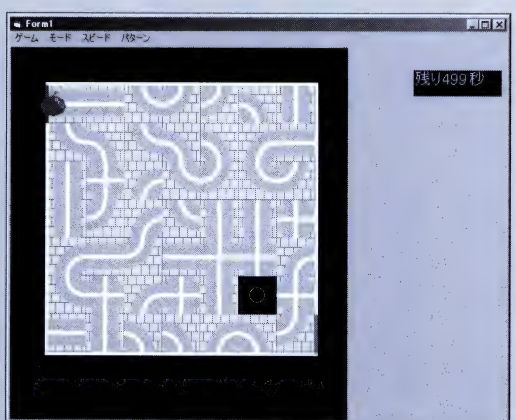
両モード共通：はみだしたり，スタート地点に戻ると即，爆発します。

ゲームスピードは3段階あり，選択できます。

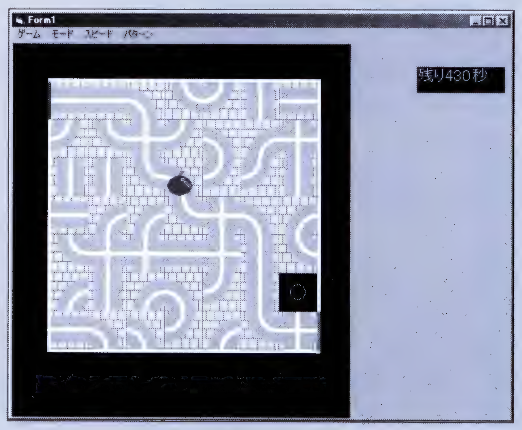
ゲームのパターン(スタート位置の違い)は全部で4パターンあり，選択できます。

操作・実行説明

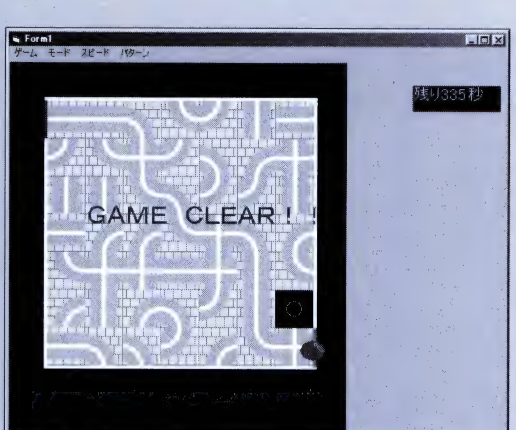
カーソルキーで○が表示されているパネル上の空白を上下左右に移動させます。



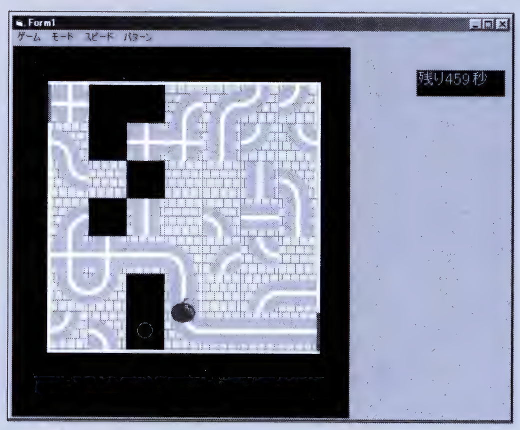
《写真1》誰でもルールを知っていそうなゲーム



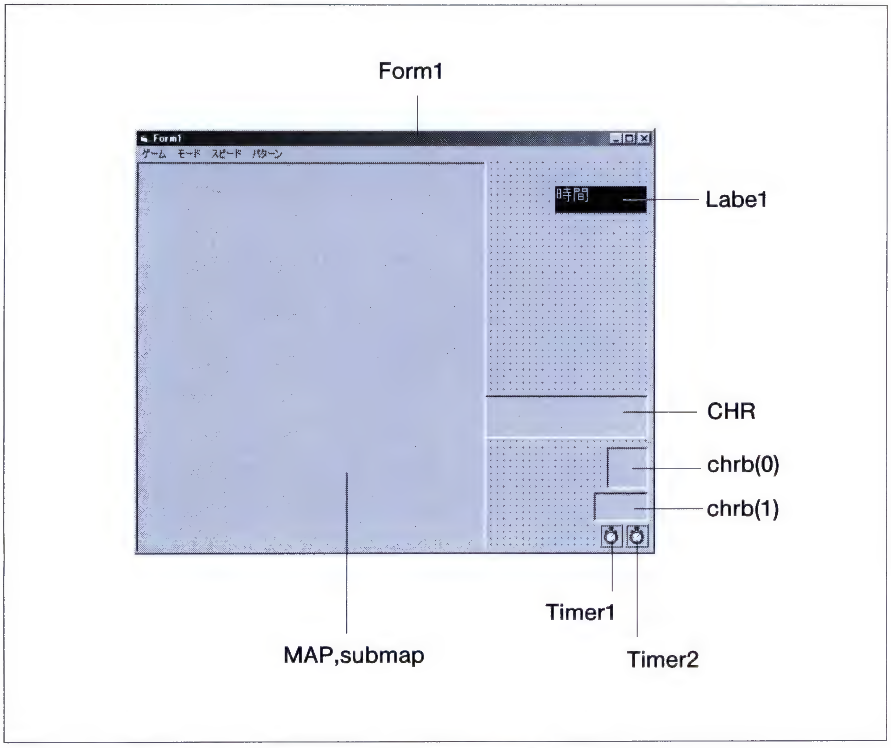
《写真2》爆弾をゴールまで誘導する道を作る



《写真3》このようにゴールまで誘導させればクリア



《写真4》アレンジを加えたオリジナルモードもある



フォームの配置図

実行には画像ファイルが必要になります。BLOCK(壁)は，QBCOLORの11番(明るいシアン)と8番(灰色)で，48×48の大きさで作ってください。DAINA（爆弾）は32×32で，マスクも32×32で作ってください。

苦労した点

初めてVisual Basicで組んだので，組み方などについては本誌1999年6月号『Five&Five』，および2000年8月号『にんにく』を解析して組んでみました。

またN88BASICとの考え方の違いなども苦労しました。

8ヶ月ぶりのプログラミングで初の言語なので…。専門誌買ったほうがいいかなぁ。

なお,改造する場合には,より速いモードなどで作ってみてください。

この作品について

この作品は以前，もう一歩で賞だった作品をオリジナルモードとパターンを加えてN88BASICから移植した作品です。

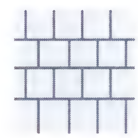


CHECKER FLAG

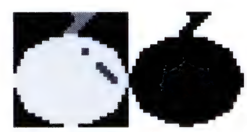
影：定番ゲームなので遊びやすい。オリジナルモードも作られているので，アイデア面でもけっして悪くない作品じゃないかな？

Dr.D：使用する画像ファイルの色が指定されている点から想像できると思うが，ブロックの道の判断を画像の色で行っているのがリスト面での特徴じゃな。画像の作成や変更などがやりにくいというデメリットがあるが，この処理のおかげでリストは短くなっている印象じゃな。

編：コントロールのプロパティ関連の処理部分を工夫すれば，かなりリストが短くできそうな感じです。



BLOCK.BMPの画像



DAINA.BMPの画像

《第2表》変数表

mx	カーソルのX座標
my	カーソルのY座標
tm	タイム(秒)
sp	ウェイト
dx	爆弾のX座標
dy	爆弾のY座標

《第1表》メニューの設定

ｷｬﾌﾟｼｮﾝ(P)	名前(M)	ｲﾝﾃﾞｯｸｽ(X)	ﾍﾙﾌﾟｺﾝﾃｷｽﾄ番号(H)	ｼｮｰﾄｶｯﾄ(S)	ﾈｺﾞｼｴｰﾄ位置(O)	ﾁｪｯｸ(C)	有効（E)	表示(V)	ｳｲﾝﾄﾞｳﾘｽﾄ(W)
ゲーム	game		0	(なし)	0－なし		○	○	
…開始	start		0	Ctrl+S	0－なし		○	○	
…終了	end		0	Ctrl+E	0－なし		○	○	
モード	mode		0	(なし)	0－なし		○	○	
…ノーマル	normal		0	(なし)	0－なし	○	○	○	
…オリジナル	ori		0	(なし)	0－なし		○	○	
スピード	speed		0	(なし)	0－なし		○	○	
…遅い	osoi		0	(なし)	0－なし		○	○	
…中間	tyuukan		0	(なし)	0－なし	○	○	○	
…速い	hayai		0	(なし)	0－なし		○	○	
パターン	pt		0	(なし)	0－なし		○	○	
…1	pt1		0	(なし)	0－なし	○	○	○	
…2	pt2		0	(なし)	0－なし		○	○	
…3	pt3		0	(なし)	0－なし		○	○	
…4	pt4		0	(なし)	0－なし		○	○	

★EXCELでBASICもWord VBAもたいして変わらないよ！！　C言語をもっと取り入れてください。『子供の科学』でVisual C++をやっている。見習ってください。多くのプログラマーを育てるために！（栃木県宇都宮市・**山ちゃん41歳**）……【**Dr.D**：ははーっ。】

《リスト》「チキチキバンバン２」のプログラム・リスト

```
Private Declare Function GetTickCount Lib "KERNEL32" () As Long
Private Declare Function BitBlt Lib "GDI32" (ByVal hdestdc As Long, ByVal x As Long,
ByVal y As Long, ByVal nwidth As Long, ByVal nheight As Long, ByVal hscrdc As Long,
ByVal xscr As Long, ByVal yscr As Long, ByVal dwrop As Long) As Long
Private Declare Function GetAsyncKeyState Lib "user32" (ByVal vkey As Long) As
Integer
Dim mp(7, 7)
Dim xx, x, y, no, co
Dim mx, my, mx1, my1, mx2, my2
Dim tm, ed, col, dp, f, dh
Dim dx, dy, dx1, dy1, dm, ddx, ddy
Dim p1 As Long, p2 As Long, p3 As Long
Dim hen As Boolean
Private Sub form_load()
   Randomize Time
   With MAP
        .AutoSize = True
        .ScaleMode = 3
        .Visible = False
        .ScaleWidth = 430
        .ScaleHeight = 480
        .Top = 1
        .Left = 1
        .AutoRedraw = True
        .BorderStyle = 0
        .BackColor = &H0
   End With
   With submap
        .AutoSize = True
        .ScaleMode = 3
        .Visible = False
        .ScaleWidth = 430
        .ScaleHeight = 480
        .Top = 1
        .Left = 1
        .AutoRedraw = True
        .BorderStyle = 0
        .BackColor = &H0
   End With
   With CHR
        .AutoSize = True
        .ScaleMode = 3
        .Visible = False
        .ScaleWidth = 600
        .ScaleHeight = 49
        .Top = 1
        .Left = 1
        .AutoRedraw = True
        .BorderStyle = 0
        .BackColor = &H0
   End With
   For r = 0 To 1
   With chrb(r)
         .AutoSize = True
         .ScaleMode = 3
         .Visible = False
         .AutoRedraw = True
   End With
```

《第3表》プロパティ表

フォームの設定	
Name	Form1
Caption	Form1
Height	7890
Left	105
LinkTopic	Form1
ScaleHeight	480
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	640
StartUpPosition	3 'Windows の既定値
Top	105
Width	9720
ピクチャボックスの設定	
Name	submap
Height	484
Left	0
ScaleHeight	480
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	430
TabIndex	5
Top	0
Width	434
Name	chrb(0)
Height	52
Index	0
Left	584
ScaleHeight	48
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	48
TabIndex	3
Top	352
Width	52
Name	chrb(1)
Height	36
Index	1
Left	568
ScaleHeight	32
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	64
TabIndex	4
Top	408
Width	68
Name	CHR
Height	53
Left	32
ScaleHeight	49
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	600
TabIndex	1
Top	288
Width	604
Name	MAP
Height	484
Left	0
ScaleHeight	480
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	430
TabIndex	0
Top	0
Width	434
ラベルの設定	
Name	Label1
BackColor	&H00000000&
Caption	時間
Font	ＭＳ Ｐゴシック
…Size	16
ForeColor	&H8000000E&
Height	33
Left	520
TabIndex	2
Top	30
Width	113
タイマーの設定	
Name	Timer1
Enabled	0 'False
Interval	1000
Left	576
Top	448
Name	Timer2
Left	608
Top	448

★日一日と寒さが増してきて，本能的に冬眠要求がおこります。冬眠前の旺盛な食欲が発生して夜は遅くまでパソコンやりながら食べつづけ，おかげでキーボードに食べカスがこびりついてしまいました。（東京都中野区・**相続人30歳**）……【**影**：PCにも夜食を？！】

```
    Next r
    chrb(0).Picture = LoadPicture(App.Path + "¥block.bmp")
    chrb(1).Picture = LoadPicture(App.Path + "¥daina.bmp")
    Label1.FontSize = 16
    Timer1.Interval = 1000
    hen = True
    chrmap
End Sub
Private Sub start_click()
    gamemain
End Sub
Private Sub end_click()
    End
End Sub
Private Sub normal_click()
    If hen = True Then
        normal.Checked = True
        ori.Checked = False
    End If
End Sub
Private Sub ori_click()
    If hen = True Then
        normal.Checked = False
        ori.Checked = True
    End If
End Sub
Private Sub osoi_click()
    If hen = True Then
        osoi.Checked = True
        tyuukan.Checked = False
        hayai.Checked = False
    End If
End Sub
Private Sub tyuukan_click()
    If hen = True Then
        osoi.Checked = False
        tyuukan.Checked = True
        hayai.Checked = False
    End If
End Sub
Private Sub hayai_click()
    If hen = True Then
        osoi.Checked = False
        tyuukan.Checked = False
        hayai.Checked = True
    End If
End Sub
Private Sub pt1_click()
    If hen = True Then
        pt1.Checked = True
        pt2.Checked = False
        pt3.Checked = False
        pt4.Checked = False
    End If
End Sub
Private Sub pt2_click()
    If hen = True Then
        pt1.Checked = False
        pt2.Checked = True
```

```
        pt3.Checked = False
        pt4.Checked = False
    End If
End Sub
Private Sub pt3_click()
    If hen = True Then
        pt1.Checked = False
        pt2.Checked = False
        pt3.Checked = True
        pt4.Checked = False
    End If
End Sub
Private Sub pt4_click()
    If hen = True Then
        pt1.Checked = False
        pt2.Checked = False
        pt3.Checked = False
        pt4.Checked = True
    End If
End Sub

Private Sub chrmap()
    '黒のＢＯＸ
    CHR.Line (0, 0)-(48, 48), QBColor(0), BF
    CHR.Circle (24, 24), 10, QBColor(13)
    '
    CHR.DrawWidth = 1
    chrbox 49
    chrbox 98
    CHR.DrawWidth = 34
    CHR.Circle (97, 48), 23, QBColor(7)
    CHR.DrawWidth = 5
    CHR.Circle (97, 48), 24, QBColor(15)
    '
    CHR.DrawWidth = 1
    chrbox 147
    chrbox 196
    CHR.DrawWidth = 34
    CHR.Circle (195, 0), 23, QBColor(7)
    CHR.DrawWidth = 5
    CHR.Circle (195, 0), 24, QBColor(15)
    '
    CHR.DrawWidth = 1
    '
    chrbox 245
    CHR.Line (252, 0)-(286, 48), QBColor(7), BF
    CHR.Line (267, 0)-(271, 48), QBColor(15), BF
    '
    chrbox 294
    CHR.Line (294, 7)-(342, 40), QBColor(7), BF
    CHR.Line (301, 0)-(334, 48), QBColor(7), BF
    CHR.Line (316, 0)-(320, 48), QBColor(15), BF
    CHR.Line (294, 22)-(342, 26), QBColor(15), BF
    '
    chrbox 343
    CHR.Line (343, 7)-(391, 40), QBColor(7), BF
    CHR.Line (343, 22)-(391, 26), QBColor(15), BF
    '
    chrbox 392
```

★毎月楽しく読ませてもらっています。最高に充実していて大好きです。できれば，私は高校生なので，パーツなどの安くていいものを手に入れる情報を教えてもらえればうれしいです。(宮崎県都城市・RYUZU16歳) ……【編：貴重な意見ありがとう。がんばります。】

```
'
   CHR.Line (441, 0)-(489, 48), QBColor(0), BF
End Sub
Private Sub chrbox(xx As Integer)
   CHR.Line (xx, 0)-(xx + 48, 48), QBColor(6), BF
   BitBlt CHR.hDC, xx, 0, 48, 48, chrb(0).hDC, 0, 0, &HCC0020
End Sub
Private Sub mawari()
   submap.Line (45, 45)-(395, 395), QBColor(14), BF
End Sub
Private Sub sg(a, b, c, d, e, f, g, h)
   submap.Line (a, b)-(c, d), QBColor(9), BF
   submap.Line (e, f)-(g, h), QBColor(12), BF
End Sub
Private Sub mapmake(a, b, c, d, e, f)
   For y = 1 To 7
    For x = 1 To 7
     mp(x, y) = 0
   Next x, y
   mp(a, b) = 7: mp(c, d) = 7: mp(e, f) = 88
   For no = 1 To 7
    For r = 1 To 7
     If no = 7 And r >= 5 Then GoTo rt
xyh:
     x = Int(Rnd(7) * 7 + 1)
     y = Int(Rnd(7) * 7 + 1)
     If mp(x, y) <> 0 Then GoTo xyh
     mp(x, y) = no
     If ori.Checked And r = 5 Then
        mp(x, y) = 33
     End If
     If ori.Checked And r = 7 Then
        mp(x, y) = 55
     End If
rt:
   Next r, no
   If ori.Checked Then
      For y = 1 To 7
       For x = 1 To 7
        If mp(x, y) = 33 Then
           mp(x, y) = 8
        End If
        If mp(x, y) = 55 Then
           mp(x, y) = 9
        End If
      Next x, y

   End If
   For y = 1 To 7
    For x = 1 To 7
     If x = e And y = f Then mp(e, f) = 0
     BitBlt submap.hDC, x * 49, y * 49, 49, 49, CHR.hDC, mp(x, y) * 49, 0, &HCC0020
   Next x, y
   mp(e, f) = 88
   bmpcopy
End Sub
Private Sub syoki()
   Timer1.Enabled = False
   If pt1.Checked Then
      sg 45, 49, 48, 97, 392, 343, 395, 392
      mx = 7: my = 7
      dx = 35: dy = 57
      ddx = 1: ddy = 1
      mapmake 1, 1, 2, 1, 7, 7
   End If
   If pt2.Checked Then
      sg 392, 49, 395, 97, 45, 343, 48, 392
      mx = 1: my = 7
      dx = 374: dy = 57
      ddx = -1: ddy = 1
      mapmake 7, 1, 6, 1, 1, 7
   End If
   If pt3.Checked Then
      sg 392, 343, 395, 392, 45, 49, 48, 97
      mx = 1: my = 1
      dx = 374: dy = 351
      ddx = -1: ddy = -1
      mapmake 7, 7, 6, 7, 1, 1
   End If
   If pt4.Checked Then
      sg 45, 343, 48, 392, 392, 49, 395, 97
      mx = 7: my = 1
      dx = 35: dy = 351
      ddx = 1: ddy = -1
      mapmake 1, 7, 2, 7, 7, 1
   End If
   dm = 1: dp = 0: dh = 0: ed = 0
   If osoi.Checked Then
      Timer2.Interval = 600
      tm = 800
   End If
   If tyuukan.Checked Then
      Timer2.Interval = 400
      tm = 600
   End If
   If hayai.Checked Then
      Timer2.Interval = 200
      tm = 400
   End If
   Label1.Caption = "残り" & tm & "秒"
End Sub
Private Sub gamemain()
   mawari
   syoki
   Timer1.Enabled = True
   Timer2.Enabled = True
   MAP.Visible = True
   hen = False
   While tm > 0 And ed = 0
    delay 100
    MAP.Cls
    keymove
    bmpcopy
    daina
    MAP.Refresh
   Wend
   Timer1.Enabled = False
   Timer2.Enabled = False
   hen = True
```

★毎月いろいろなことが載っているので，とても勉強になってます。（新潟県長岡市・**南　寿和19歳**）……【**編**：ありがとうございます。こういう声を励みに今年もがんばります！今，みなさんが知りたいこと，興味を持っていること，編集部までお知らせください。】

```
    If ed <> 0 Or tm = 0 Then gameover
  End Sub
  Private Sub keymove()
    mx1 = (GetAsyncKeyState(&H25) <> 0 And mx > 1) - (GetAsyncKeyState(&H27) <> 0 And
  mx < 7)
    my1 = (GetAsyncKeyState(&H26) <> 0 And my > 1) - (GetAsyncKeyState(&H28) <> 0 And
  my < 7)
    mx2 = mx: my2 = my
    mx = mx + mx1: my = my + my1
    If mx <> mx2 Or my <> my2 Then
      mymove
    End If
  End Sub
  Private Sub mymove()
    mp(mx2, my2) = mp(mx, my)
    BitBlt submap.hDC, mx2 * 49, my2 * 49, 49, 49, CHR.hDC, mp(mx2, my2) * 49, 0,
  &HCC0020
    mp(mx, my) = 88
    BitBlt submap.hDC, mx * 49, my * 49, 49, 49, CHR.hDC, 0, 0, &HCC0020
  End Sub
  Private Sub bmpcopy()
    BitBlt MAP.hDC, 45, 45, 400, 400, submap.hDC, 45, 45, &HCC0020
  End Sub
  Function delay(ByVal tim As Long)
    Dim t_1 As Long, t_2 As Long
    t_1 = GetTickCount
    t_2 = t_1
    While ((t_2 - t_1) < tim)
     t_2 = GetTickCount
     DoEvents
    Wend
  End Function
  Sub timer1_timer()
    tm = tm - 1
    Label1.Caption = "残り" & tm & "秒"
  End Sub
  Sub timer2_timer()
     dainamove
  End Sub
  Private Sub daina()
    BitBlt MAP.hDC, dx, dy, 32, 32, chrb(1).hDC, 32, 0, &H8800C6
    BitBlt MAP.hDC, dx, dy, 32, 32, chrb(1).hDC, 0, 0, &HEE0086
  End Sub
  Private Sub dainamove()
    dx1 = dx: dy1 = dy: f = 0
  dn:
    If dm = 1 Then
      dx = dx + ddx
      p1 = submap.Point(dx + 16, dy + 16)
      p2 = submap.Point(dx1 + 16, dy + 13)
      p3 = submap.Point(dx1 + 16, dy + 19)
      If p1 = QBColor(15) Then dp = 0: GoTo dow
      hantei
      If f = 1 Then dx = dx1: GoTo dow
      If p2 = QBColor(15) Then dx = dx1: dm = 2: ddy = -1: GoTo dn
      If p3 = QBColor(15) Then dx = dx1: dm = 2: ddy = 1: GoTo dn
    End If
    If dm = 2 Then
      dy = dy + ddy
      p1 = submap.Point(dx + 16, dy + 16)
      p2 = submap.Point(dx + 13, dy1 + 16)
      p3 = submap.Point(dx + 19, dy1 + 16)
      If p1 = QBColor(15) Then dp = 0: GoTo dow
      hantei
      If f = 1 Then dy = dy1: GoTo dow
      If p2 = QBColor(15) Then dy = dy1: dm = 1: ddx = -1: GoTo dn
      If p3 = QBColor(15) Then dy = dy1: dm = 1: ddx = 1: GoTo dn
    End If
  dow:
  End Sub
  Private Sub hantei()
    If normal.Checked Then
      If p1 = 0 Then dp = dp + 1
      If p1 = QBColor(11) Or p1 = QBColor(8) Or p1 = QBColor(6) Then dp = dp + 1
      If p1 = QBColor(12) Then ed = 1
    End If
    If ori.Checked Then
      If p1 = 0 Then ed = 2
      If p1 = QBColor(12) Then
        If dh = 1 Then ed = 1
        If dh = 0 Then ed = 2
      End If
      If p1 = QBColor(11) Or p1 = QBColor(8) Or p1 = QBColor(6) Then
        ddx = -ddx: ddy = -ddy
        dx = dx1: dy = dy1
        dainamove
        f = 1: dh = 1
      End If
    End If
    If p1 = QBColor(14) Then ed = 2
    If p1 = QBColor(9) Then ed = 2
    If dp >= 18 Then ed = 2
    If ed <> 0 Or dp <> 0 Then f = 1
  End Sub
  Private Sub gameover()
    If ed = 2 Then
      bmpcopy
      MAP.FillStyle = 0
      MAP.FillColor = QBColor(4)
      MAP.Circle (dx + 16, dy + 16), 25, QBColor(4)
      MAP.FillColor = QBColor(6)
      MAP.Circle (dx + 16, dy + 16), 10, QBColor(6)
      MAP.FillStyle = 1
    End If
    col = 0
    While 1
     MAP.CurrentX = 100: MAP.CurrentY = 180: MAP.FontSize = 25
     MAP.ForeColor = QBColor(col)
     If ed = 1 Then
       MAP.Print "ＧＡＭＥ　ＣＬＥＡＲ！！"
       Else
       MAP.Print "ＧＡＭＥ　ＯＶＥＲ！！"
     End If
     MAP.Refresh
     delay 200
     col = col + 1
     If col > 15 Then col = 0
     DoEvents
    Wend
  End Sub
```

★年の瀬に『ベーマガ』扱ってる本屋がつぶれてしまった。どうしよう……。ただでさえ見つからないのに…。（青森県北津軽郡・田舎者20歳）……**【販売部長：よく行く本屋にベーマガがない！などの情報は販売部にご連絡を。置いてもらうよう働きかけますので。】**

リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net)からダウンロードしてお楽しみください。

Windows(Visual Basic V6.0)用



ホームページのテーブル作成にどうぞ

HTMLテーブルメーカー

山澤正裕（30歳）

入力方法は32ページをご覧ください。

内容

このプログラムは，ホームページ作成で使われるHTMLのテーブル(表組み)のコードを,画面上での簡単な操作で作成するツールです。動作環境は，ディスプレイが800×600ドット以上必須とさせていただきます。

プログラム実行時の注意

このプログラムは，実行時にブラウザでプレビューするための一時的なHTMLファイル(temporary.html)を本プロジェクトと同一ディレクトリに書き込みます。よって，このプログラムを書き込み可能ドライブに保存した上で，使用してください（ハードディスクへの保存が良い)。

コンパイルした実行ファイル(htmlTableMaker.exe)を別の場所に移動しての使用や，実行ファイル単体だけでの使用も可能ですが，プロジェクトから実行するのと同様に，実行ファイルは実行時に自分自身と同一ディレクトリに一時ファイルを作成します。その点に注意する必要があります。なお，作成されたHTMLファイルは，プログラム実行後には削除してもかまいません。

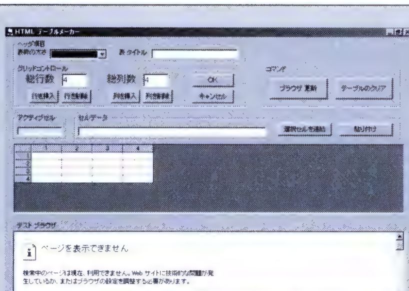
使用方法

ヘッダ項目は，入力を省略することが可能です。入力すれば，表に枠をつけたり，見出しをつける事ができます。

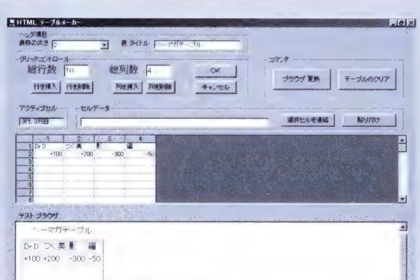
グリッドコントロール欄で，作成する表の行と列の管理を行います。表の総行数，総列数を入力したら，OKボタンを押してください。設定値の上限は，行数=80，列数=40とします。それ以上の大きな表は,Webに適さないと判断しました。

画面中央のグリッドで，HTML化したい表を入力します（写真2)。入力したいセルをクリックすると，そのセルが選択状態になります。同時に複数のセルを選択することも可能です。そこでEnterキーを押すか，ダブルクリックすることで，セルデータ欄にフォーカスが移ります。セルデータ欄のテキストボックスに文字を入力して，Enterキーを押すと，グリッドの選択されたセルにデータが入ります。

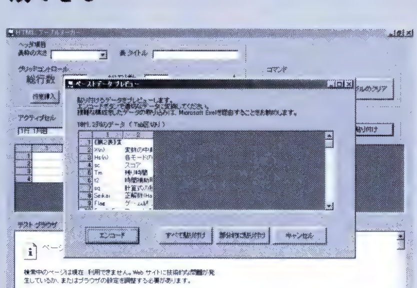
コマンド欄のブラウザ更新ボタンで，画面中央のグリッドで表現したデータを，



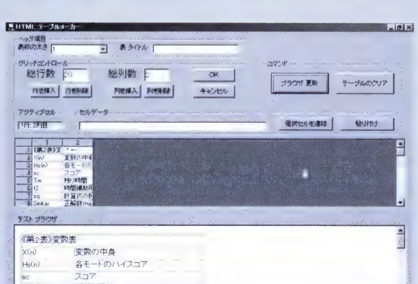
《写真１》HTMLのテーブルが手軽に作成できる



《写真２》ブラウザ機能があるので，出来具合がすぐわかる



《写真３》また，クリップボードにあるテキストデータを



《写真４》簡単な作業でHTMLのテーブルにできる

CHECKER FLAG

つぐ美：ソースを載せてっていうお願いがあったのですが，リスト容量の合計が31KBでは，さすがに無理でした。ごめんなさい。何回か言っていますが，リストの上限は15KBを目安にお願いします。また，VBの場合はフォームのコントロールの数にも気をつけてくださいね。

Dr.D：画面構成やテーブル作成に関する細かな指定など，ファイル作成ソフトとしての基本はしっかり持っている。こういった実用ソフト制作に興味がある読者は「Piku Up LIST」で取り上げているリストだけでなく，ダウンロードファイルにあるリストにも目を通して見るのがオススメじゃな。

編：ブラウザ機能を簡単に実現している部分に，VBプログラムのメリットが見えますね。

★はじめまして。知り合いがつぎつぎブロードバンド化しているので，ADLS申し込んだら電話局が遠くて無理とのこと。常時接続の道は遠い…。（千葉県市原市・ゆう@26歳）……【編：ちなみに私の自宅でのインターネット環境は，CATVのHomePNA方式です。】

画面下側のテストブラウザでウェブページとして再現します。このブラウザは，本物の機能を持ったブラウザです。

テストブラウザでの再現で，満足するテーブルができたら，ブラウザのエリア内で右クリックしてください。右クリックのメニューから「ソースの表示」を選んで，ソースコードを表示します。あとは，表示されたコードをコピーして，ご自由に使ってください。

あと，便利だと思われる機能を以下に２つ紹介します。

その1：セルの連結

選択したセルを，選択セルを連結ボタンで，セルを連結したデータとして表現することができます。連結エリアには，連結データの表現方法として以下のキーが表示されます。

<!←>：このセルは，左隣のセルに連結します。

<!↑>：このセルは，上方のセルに連結します。

<!□>：このセルは，連結セルの範囲を表現するためのダミーセルです。

一応，表現方法に意味があるのですが，ユーザーには，表現がわかりにくいかもしれません。でも，そういうものだと簡単に解釈して使ってもらえると，ありがたいです。テストブラウザで簡単にプレビューできるから，そんなに困ることはないとは思っています。

その2：クリップボードからの貼り付け

貼り付けボタンで，クリップボードのデータをグリッドへ貼り付けることができます。貼り付けボタンを押すと，クリップボードのデータをプレビューするフォームが表示されます（**写真3**）。エンコードボタンを押すことで，クリップボードのデータの区切り方を選択することができます。データの区切り方を決めたら，すべて貼り付けボタン，または範囲を選択して部分的に貼り付けボタンを押します。

他のアプリケーションからのデータで取り込んで，HTMLのテーブルを作成できるところが，このツールの一番便利な機能だと思います。

おまけのアドバイス

貼り付けたいデータが複雑で，きれいにデータを区切れないときは，対象のデータを一度,Microsoft Excelのシートに貼り付けて，そこからコピーしてみてください。そうするだけで，きれいにセルを分けてくれる場合があります。

最後に

本誌2001年7月号の『MY電卓』に続いての投稿です。

今回のプログラムは，肝心な機能の部分は丁寧に作りこみ，おまけのような細かな処理はなるべく抑えたつもりです。あまり，おまけ要素を加えすぎると，大切な本質の部分が見えにくくなってしまうので，ソリッドなものにしようと心がけました。それでも，結局ボリュームが大きくなりました。

前回の『MY電卓』は，「個性に欠ける」という編集部の評価でしたが，今回はどうでしょうか？私のメールアドレスは，yamazawa@uoz.fitweb.or.jpです。メールによるご感想をお待ちしています。

Pick Up LIST

《リスト１》HTMLコード作成部分のリスト

```
    'HTMLコード記述
    If cboBorder.Text = "" Then
        vCode = "<TABLE>" & vbCrLf
    Else
        vCode = "<TABLE border=" _
         & Chr(34) & cboBorder.ListIndex & Chr(34) & ">" & vbCrLf
    End If

(途中省略)

' 一時ファイルにHTMLコードを書き込むサブルーチン
Private Sub OutputTo_TmpFile(vHtmlCode As String)
    Dim i As Integer, vCodeArr() As String

    vCodeArr = Split(vHtmlCode, vbCrLf) '改行コードを区切りに、配列に格納。

    '書き込みモードでオープン
    Open "temporary.html" For Output As #1 'ファイル番号 1

    '書き込み
    Print #1, "<HTML><BODY>"
    Print #1, "" '空行

    For i = 0 To UBound(vCodeArr)
        Print #1, vCodeArr(i)
    Next i

    Print #1, "" '空行
    Print #1, "</BODY></HTML>"

    Close #1

End Sub
```

Dr.Dのワンポイント解説

Dr.D：当たり前の事じゃが，この作品はファイルを作り出すソフトなので，ここで取り上げているファイル作成処理が必要になってくる。このソフトはHTMLファイルを作成するので，テキスト形式でHTMLのタグを記述していくことになる。書き込むタグは<HTML>タグといった初期部分を除いては，String型の配列にテーブル作成に関するタグを持っておき，それをFor文内ですべて書き込みする。つまり，このソフトにおいて，このファイル処理を除く主な処理は，タグを保存しておくString型の配列作成に関連してくるというわけじゃな。

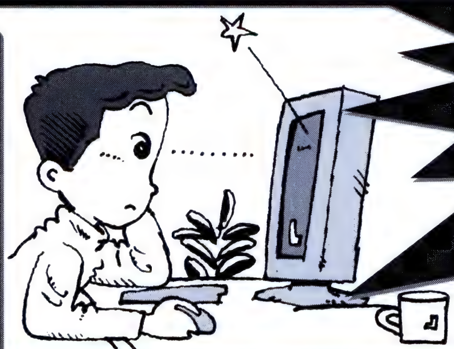
★インターネット読者ハガキをはじめて知りました。切手要らずでいいですね。（大阪府茨木市・J-1 29歳）……【編：ハガキとインターネットでの投稿，載る確率は？という質問を受けますが，どちらも同じです。お好きなほうで参加してください。よろしく！】

入力方法は32ページをご覧ください。

Windows(Visual Basic V6.0)用

アイデア勝負のスクリーンセーバー

Light



清家　巧（17歳）

内容

Windowsのデスクトップ画像を利用して実行するスクリーンセーバーです。

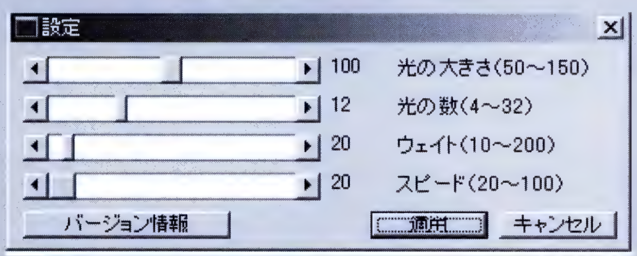
使い方

リストを入力した後,ファイルメニューからEXEファイルを作成し、拡張子(.exe)を.scrに変更します。

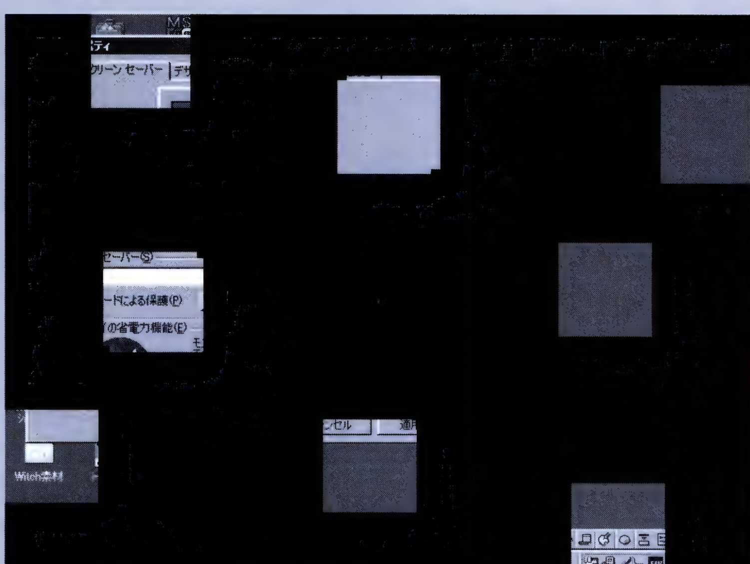
変更したら、他のscrファイルがあるフォルダ(windows￥system32など）へ作成したファイルをコピーし、windowsの画面プロパティからLightをスクリンセーバーとして選択すれば、実行できます。

この作品の機能は、スクリンーセーバーの役割である画面やけを防ぐこと,もしくはアクセサリーとしての役割もあるかもしれません。

ウェイトを下げるとスピードが上がり,描画間隔が短くなるので，ウェイトを小さくする場合，スピードを落とすほうが無難です。



《写真２》画面効果に関する細かな設定も可能



《写真１》デスクトップ画面を部分的に表示することで，ライト効果っぽく見せる

CHECKER FLAG

影：ベーマガ投稿プログラムでは，なぜか数が少ないスクリーンセーバー作品。知っている人も多いだろうけど，スクリーンセーバーは基本的には実行ファイル(exeファイル)と同じ。なので，VBなどでも制作は可能なのだ。

つぐ美：ただ，チョット物足りないというのが正直な感想ね。基本的な構成が出来ているのだから，ここからもっともっと改良を重ねてほしいわね。

編：例えばライトのように見せるなら，表示する画像の色を変えて色のついたスポットライトっぽく見えることができそうですね。他にもライトの形を四角以外でも表現してほしいですね。マスク処理などを使えば実現できるのでは？

Dr.D：比較的短いリストで制作されているのが長所と言えるな。設定画面などもしっかり作られているので，スクリーンセーバープログラムの入門用としては，この作品のリストは参考になるはずじゃ。

★こんな俺もとうとう成人式です！『ベーマガ』とともに歩んだ10代後半でした。これからもよろしくです。（東京都中央区・まぢけん20歳）……【つぐ美：これからが人生大変よ。影：あれ？20歳のはずのつぐ美ちゃんが，ずいぶん年を重ねたようなセリフを！】

ウェイトは短いほうがきれいにアニメーションします。光の数は12個くらいが結構きれいです。

操作方法

スクリーンセーバーなので，実行された画面を眺めることになります。マウスを動かしたりキーボードを叩くと終了します。キーボードを激しく叩きすぎると壊れます。

プログラムの仕組み

デスクトップ画像をAPIを利用して非表示のPictureboxに格納して，フォームにその画像を転送してスクリーンセーバーにしています。

なお，フォームを最前面に表示するようにしてあるので多分スクリーンセーバーを邪魔するウインドウは入りません。

入力方法

light.srmは,ピクチャーボックスを貼り付け対応しているプロパティ(**第2表**)を設定して，あとは**リスト1**をそのまま入力すれば多分大丈夫です。light.frmのMouseIconで指定するcur.icoは，何も表示しないアイコン画像になります。

設定のフォームのほうは，プロパティ表(**第1表**)などを参考に移せばいいだけだと思います。

苦労した点

ある2ヶ所の変数をRedimで宣言するのが，まだ良くわかんないんですよねぇ。むしろ何か意味があったのかも,よくわかんないです。でもDimだとエラーがでるんですよ…。

改造するなら…？

起動ごとに変えようと努力はしたのですが，気のせいか同じ動きばかりします。また，起動中も同じ動きの繰り返しなので，少し変わった動きをさせるとか…。

《第1表》light設定.frmのプロパティ表

フォームの設定	
Name	Form2
BorderStyle	1 '固定(実線)
Caption	設定
Height	2265
Icon	(ｱｲｺﾝ)
Left	0
LinkTopic	Form2
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	1890
ScaleWidth	5745
StartUpPosition	3 'Windows の既定値
Top	0
Width	5835
コマンドボタンの設定	
Name	Command1
Caption	バージョン情報
Height	255
Left	120
TabIndex	14
Top	1560
Width	1935
Name	Command2
Caption	キャンセル
Height	255
Left	4560
TabIndex	4
Top	1560
Width	1095
Name	tekiyou
Caption	適用
Height	255
Left	3360
TabIndex	3
Top	1560
Width	1095
ラベルの設定	
Name	LABsize
Height	255
Left	3000
TabIndex	13
Top	120
Width	495
Name	LABkazu
Height	255
Left	3000
TabIndex	12
Top	480
Width	495
Name	LABwait
Height	255
Left	3000
TabIndex	11
Top	840
Width	495
Name	LABspeed
Height	255
Left	3000
TabIndex	10
Top	1200
Width	495
Name	Label1
Caption	光の大きさ（50～150）
Font	ＭＳ Ｐゴシック
…Size	10
Height	255
Left	3600
TabIndex	0
Top	120
Width	2055
Name	Label2
Caption	光の数（4～32）
Font	ＭＳ Ｐゴシック
…Size	10
Height	255
Left	3600
TabIndex	1
Top	480
Width	1695
Name	Label3
Caption	ウェイト（10～200）
Font	ＭＳ Ｐゴシック
…Size	10
Height	255
Left	3600
TabIndex	2
Top	840
Width	1695
Name	Label4
Caption	スピード（20～100）
Font	ＭＳ Ｐゴシック
…Size	10
Height	255
Left	3600
TabIndex	5
Top	1200
Width	1695
水平スクロールバーの設定	
Name	speed
Height	255
LargeChange	10
Left	120
Max	100
Min	20
TabIndex	9
Top	1200
Value	50
Width	2775
Name	wait
Height	255
LargeChange	10
Left	120
Max	200
Min	10
TabIndex	8
Top	840
Value	20
Width	2775
Name	kazu
Height	255
Left	120
Max	31
Min	3
TabIndex	7
Top	480
Value	3
Width	2775
Name	size
Height	255
LargeChange	10
Left	120
Max	150
Min	50
TabIndex	6
Top	120
Value	50
Width	2775

★昔（'95～'97年）のベーマガは付録がついていたが，今はついていない。復活を求む！できればCDなどを。（埼玉県大里郡・**mass17歳**）…
…【**編**：復活のおまじないを！そーれ，そーれ，そーれ，そーれ！**つぐ美**：あっ！二日酔いの影さんがいま復活しました。】

《リスト１》フォーム１(light.frm)のプログラム・リスト

```
Option Explicit
'正確な経過時間を調べる
Private Declare Function GetTickCount Lib "kernel32" () As Long
'デスクトップのウインドウハンドル取得する。
Private Declare Function GetDesktopWindow Lib "user32" () As Long
'ウインドウ画像のデバイスコンテキスト取得する。
Private Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long
'デバイスコンテキストの解放する。
Private Declare Function ReleaseDC Lib "user32" (ByVal hwnd As Long, ByVal hDC As Long) As Long
'描画用
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As Long, ByVal Y As
Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long,
ByVal ySrc As Long, ByVal dwRop As Long) As Long
'前面に表示用
Private Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal hWndInsertAfter
As Long, ByVal x As Long, ByVal Y As Long, ByVal cx As Long, ByVal cy As Long, ByVal wFlags As
Long) As Long

Dim xxx As Integer, yyy As Integer '動き回る範囲
Dim asd As Boolean 'ループのフラグ
Dim MX As Single, MY As Single, MK As Byte 'マウスが動いたか判定用
Const rg = 3.1415926 / 180
Dim kazu As Integer '（この値+1）個の光ができる！
Dim size As Integer '光の大きさ
Dim wait As Integer '1コマの時間(□秒)
Dim speed As Integer '光の移動スピード
Private Function Set_FormTop(frmObject As Form, Optn As Integer) As Long

    Dim objHwnd As Long

    objHwnd = frmObject.hwnd

    '手前に表示にセット
    Set_FormTop = SetWindowPos(objHwnd, -1, 0, 0, 0, 0, _
                        &H40 Or &H2 Or &H1)

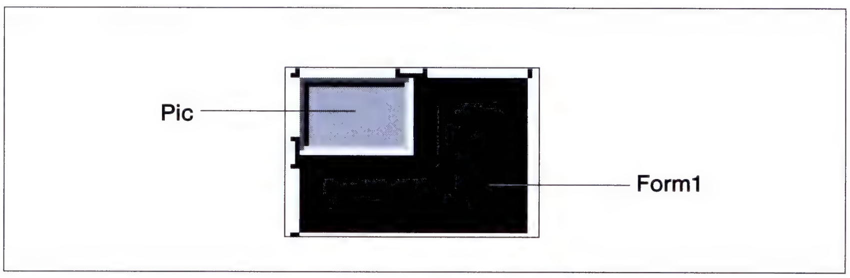
End Function
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
asd = False 'ループを抜け
End Sub

Private Sub Form_Load() '初期設定
Dim c As String
c = Left$(Command, 2)              '実行判定
If App.PrevInstance = True Or c$ = "/P" Or c$ = "/p" Or c$ = "/A" Or c$ = "/a" Then End: Exit
Sub '二重起動禁止！
    If c$ <> "/S" And c$ <> "/s" Then
        Load Form2
        Form2.Show
        Unload Form1
        Exit Sub
    End If
asd = True 'ループのフラグを立てる

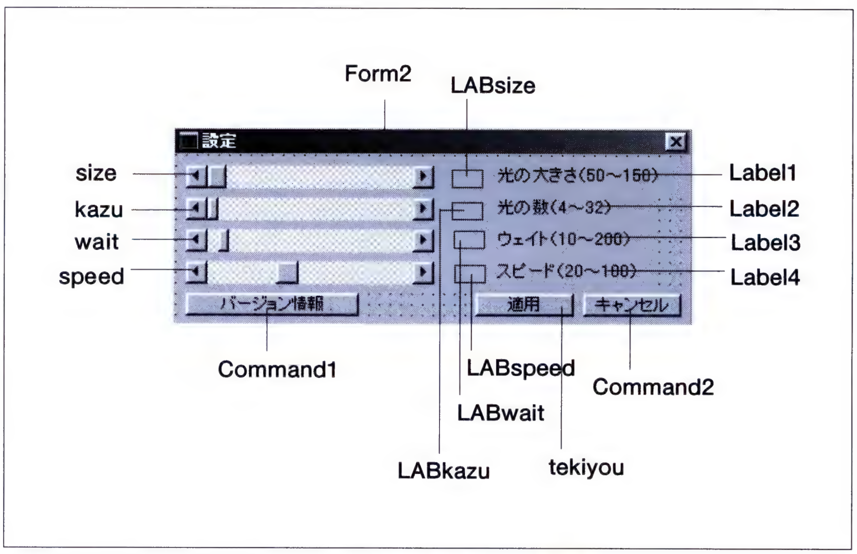
kazu = GetSetting("SimaLaboratory", "Light", "kazu", 11) '（この値+1）個の光ができる！ただし
kazu>=1でないといけない
size = GetSetting("SimaLaboratory", "Light", "size", 100) '光の大きさ
```

《第2表》light.frmのプロパティ表

フォームのプロパティ表	
Name	Form1
AutoRedraw	-1 'True
BackColor	&H80000006&
BorderStyle	0 'なし
Caption	Form1
Height	510
Icon	(ｱｲｺﾝ)
Left	0
LinkTopic	Form1
MouseIcon	cur.ico
MousePointer	99 'ﾕｰｻﾞｰ定義
ScaleHeight	34
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	50
ShowInTaskbar	0 'False
StartUpPosition	3 'Windows の既定値
Top	0
Width	750
WindowState	2 '最大化
ピクチャボックスの設定	
Name	Pic
AutoRedraw	-1 'True
Height	17
Left	0
ScaleHeight	13
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	21
TabIndex	0
Top	0
Visible	0 'False
Width	25



《フォーム1》light.frmの配置図



《フォーム2》light設定frmの配置図

★最近Visual Basicで3D modelerを2〜4時間／日で作っています。使い勝手がよくなるようにがんばっています。そのおかげで友人から変態呼ばわりされた。プログラムってそんなに変態行為じゃないのに（シクシク）。（岩手県岩手郡・**佐藤祐司17歳**）……【影：ねぇ。】

```
wait = GetSetting("SimaLaboratory", "Light", "wait", 20) '1コマの時間(□ 秒)
speed = GetSetting("SimaLaboratory", "Light", "speed", 20) '光の移動スピード

Form1.Show 'フォームを表示

With Pic
    .Width = Form1.ScaleWidth
    .Height = Form1.ScaleHeight
End With
'デスクトップの画像をpicに格納する
Dim Desk As Long
Dim DeskTop As Long

    Desk = GetDesktopWindow
    DeskTop = GetDC(Desk)

Call BitBlt(Pic.hDC, 0, 0, Form1.ScaleWidth, Form1.ScaleHeight, _
            DeskTop, 0, 0, &HCC0020)
Pic.Refresh
'解放する
Call ReleaseDC(Desk, DeskTop)

'動き回る範囲を決める
xxx = Form1.ScaleWidth - size
yyy = Form1.ScaleHeight - size

Dim ret As Long '手前に表示にする。(邪魔なウインドウを後ろにする。)
    ret = Set_FormTop(Form1, -1)

Call main
End Sub

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, x As Single, Y As Single)
asd = False 'ループを抜ける
End Sub
Sub main()
ReDim lightx(kazu) As Integer
ReDim lighty(kazu) As Integer 'ライトの場所を入れる変数
Dim nt As Long, lt As Long '経過時間を出すのに使う
Dim nami As Double '動きに使う。
Dim i As Integer 'For…Nextで使う。
Dim q As Integer
q = Int(Rnd(0) * 2) + 1 '動きを変えるため(起動のたびに)
Do While asd = True
    nt = GetTickCount() '現在の時間を格納
    If nt - lt >= wait Then 'wait□ 秒以上経過時
    Form1.Cls '消す
        nami = nami + (speed / 100)
        For i = 0 To kazu Step 2 'ライトの動き(2パターン)また、iに足す数値(整数でなくても良い)
で動きが変わる。
            lightx(i) = Sin(nami * (i + q + 1) * rg) * xxx / 2 + xxx / 2
            lighty(i) = Cos(nami * (i + q + 2) * rg) * yyy / 2 + yyy / 2
        Call BitBlt(Form1.hDC, lightx(i), lighty(i), size, size, Pic.hDC, lightx(i),
lighty(i), &HCC0020)
        Next
        For i = 1 To kazu Step 2
            lightx(i) = Cos(nami * (i + q + 1) * rg) * xxx / 2 + xxx / 2
            lighty(i) = Sin(nami * (i + q + 2) * rg) * yyy / 2 + yyy / 2
        Call BitBlt(Form1.hDC, lightx(i), lighty(i), size, size, Pic.hDC, lightx(i),
lighty(i), &HCC0020)
        Next
        lt = GetTickCount() '最終処理時の時間を格納
        Form1.Refresh
    End If
    DoEvents
Loop
Unload Me
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, x As Single, Y As Single)
If MK = 0 Then '動かしたら終了。
MX = x: MY = Y
MK = 1
ElseIf MK = 1 Then
    If MX <> x Or MY <> Y Then asd = False 'ループを抜ける
End If
End Sub
```

《リスト２》フォーム２(light設定.frm)のプログラムリスト

```
Private Sub Command1_Click()
Dim c As String
c = MsgBox("LIGHT" & Chr(13) & Chr(13) & "製作者　清家　巧", 64, "LIGHT")
End Sub

Private Sub Command2_Click()
Unload Me
End Sub

Private Sub Form_Load()
size.Value = GetSetting("SimaLaboratory", "LIGHT", "size", 100)
kazu.Value = GetSetting("SimaLaboratory", "LIGHT", "kazu", 11)
wait.Value = GetSetting("SimaLaboratory", "LIGHT", "wait", 20)
speed.Value = GetSetting("SimaLaboratory", "LIGHT", "speed", 20)
End Sub

Private Sub kazu_Change()
LABkazu.Caption = kazu.Value + 1 '修正
End Sub
Private Sub Size_Change()
LABsize.Caption = size.Value
End Sub
Private Sub speed_Change()
LABspeed.Caption = speed.Value
End Sub
Private Sub wait_Change()
LABwait.Caption = wait.Value
End Sub

Private Sub tekiyou_Click()
SaveSetting "SimaLaboratory", "Light", "kazu", kazu.Value
SaveSetting "SimaLaboratory", "Light", "speed", speed.Value
SaveSetting "SimaLaboratory", "Light", "size", size.Value
SaveSetting "SimaLaboratory", "Light", "wait", wait.Value
MsgBox "設定を有効にしました。"
Unload Me
End Sub
```

★ウイングアローが名古屋優駿で先頭にたったときの実況が「ジンガロ」と聞こえた。(千葉県山武郡・**野潤上15歳**)……【**影**：15歳にして，競馬を語れるとは！**編**：「ダビスタ」の話では？**影**：なんにしても，私は負けない！今年はがんばる！**編**：ほっときましょ。】

Windows(Visual Basic V5.0)用



変数の値を暗記できるか？

あーる・かるきゅれーしょん Ver1.1

ていこく（22歳）

入力方法は32ページをご覧ください。

ゲームの内容

基本的には暗算ですが，記憶力と判断能力も必要とするゲームです。タイトルの「あーる」はReplace(置き換え)の頭文字です。

入力方法

①VBを起動し，フォームにオブジェクトを貼り付けます。
②フォームのリスト（**リスト1**），モジュールのリスト（**リスト2**）をそれぞれ入力します。
③VBのメニューから「プロジェクト」→「Project1のプロパティ」を選択し，「全般」の「スタートアップの設定」を「Sub Main」に変更して，保存します。
④一度VBを終了し，再び起動し読み込んで，実行してください。

最初はプログラム中で「'/*」と「'*/」に挟まれた部分は打ち込まないで，エラーが出なくなるまで実行してチェックしてください。エラーが出なくなったと思ったら「'/*」～「'*/」の部分を追加してください。

遊び方

プログラムを実行するとタイトル画面が表示されます(**写真1**)。そこで「Easy」「Normal」「Hard」のいずれかを選択するとゲームスタートです。

まず，[A][B][C]の三つの変数とその中身が表示されるので，それを覚えます(**写真2**)。３秒経つと，表示が消えて，メイン画面に移動します。

メイン画面では計算式(足し算，引き算，掛け算)が次々出てくるので，答えを右のテキストボックスに入力し，下の「回答」ボタンを押してください（**写真3**）。

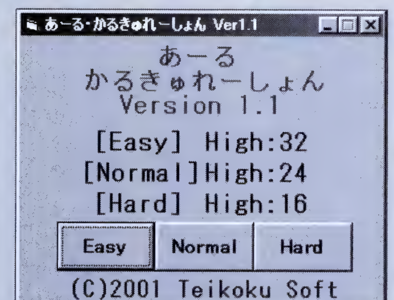
計算式には“[A]+1=?”といった感じで，[A][B][C]が混ざった式も出てきます。この場合は，それぞれに最初に覚えた数値を当てはめて計算してください。

正解なら得点が加算され，不正解なら正解を表示します。加算される得点は，変数の無い式が１点，変数が１つ混じっている式が２点，変数が二つ混じっている式が３点となります。

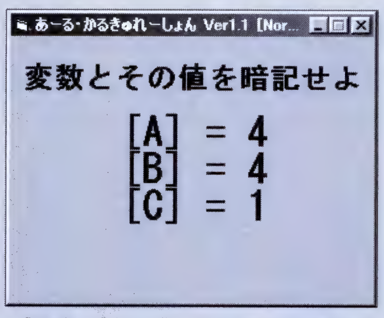
制限時間は90秒で，スコアを競います。ハイスコアはファイルに記録されていきます（**写真4**）。

Normalは，基本的なルールはEasyと同じですが，３問正解するごとに，[A][B][C]のいずれか一つの変数の中身が変わっていくので難しくなります。

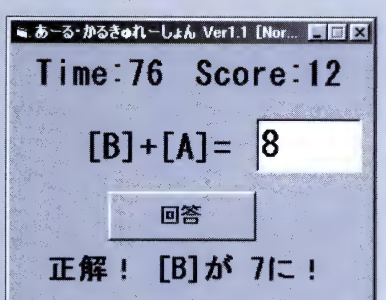
Hardは計算式が三項式(例:1+2+3)になり，Normalと同じく３問正解するたびに変数の中身が変わっていくのでさら



《写真１》計算の練習にはうってつけのソフト？



《写真２》まず，この三つの変数の値を暗記して

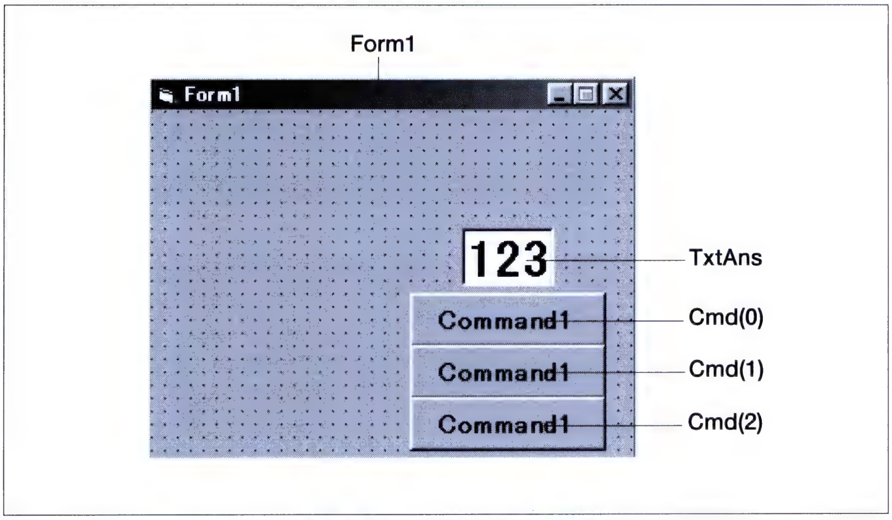


《写真３》計算式に答えていく。変数の値も変化するゾ



《写真４》制限時間終了後に得点を表示

★PS版『ドラクエⅣ』の最大のメリットは，データが消えないことだと個人的には思う。（神奈川県横浜市・**バビブ改めボーちゃん20歳**）……【**編**：そう言えば，影さん，ドラクエやってます？**影**：いやー，なかなか仕事が忙しくてね。**編**：え？仕事？し・ご・と？】



《第1図》フォームの配置図

改造のポイント

に難しくなります。

制限時間を増やすには，GameInit内の「Tm = 90」の90を大きくします。計算式の種類やゲーム中に出てくる変数の数を増やしてみるといいでしょう。

あとは絵を追加して演出をしてみたり，ノルマを設定して面クリア型にしてみたり……。

作成後記

計算ゲームという,ありきたりなタイプですが,「変数」を使ったことで少しはオリジナリティをもたせられたのではないかと思います。

友人にテストプレイをしてもらったところ，「答えて次の問題に出たとき，カーソルをテキストボックスに移動させるのが面倒」と、ユーザーインターフェースの悪さを指摘されたので，SetFocusを使うことで，次の問題が出たとき，テキストボックスに自動的にカーソルが移動されるようにしました。

何ヶ所かに見られる"While (Z=-1)：～：DoEvents：Wend"というのは，変数Zが1である限り，先の処理には進まないことを意味しています。今回の場合，変数Zが変わるのはコマンドボタンが押されたときですので，つまりは「コマンドボタンが押されるまで次には進まない」ということになります。結構便利な使い方かもしれませんので，コマンドボタンを使ったゲームを作るときに，興味があったら使ってみてはいかがでしょうか？

実はこのプログラム，本誌2001年12月号の一口CHECKER FLAGに掲載されたプログラムを改良したものです。そこでは，つぐ美さんから「問題の式の形が，もう少しバラエティに富んでいるといいんじゃないかしら？」というご指摘を受けました。以前投稿したバージョンでは，問題の式はすべて二項式でしたが，それでは割り算を除けば問題の式の形は３種類しかありませんでした。そこで，新たに設けたHardレベルでは出題される式を三項式にし，式の形を8種類に増やしました（本当は9種類なのですが，A*B*Cのような掛け算を２回行う式は暗算が大変なので除外しています)。

あ，変数の中身をメモにとりながらゲームをするのは反則ですよ（笑）。各レベルの私の最高点は，Easyが82点，Normalが82点，Hardが39点です。

私のメールアドレスはymd3106@jomon.ne.jp，HPアドレスはhttp://www.jomon.ne.jp/~ymd3106/です。感想お待ちしています。HPにも遊びに来てくださいね。

《第2表》変数表

X(n)	変数の中身
Hs(n)	各モードのハイスコア
sc	スコア
Tm	残り時間
t2	時間補助用(t2=10のときTm-1)
sq	計算式の種類
Seikai	正解数(Hardモードで使用)
Flag	ゲーム終了のフラグ
FieldX,FieldY	フォームの幅と高さ
Title	タイトル名
fnum	ファイルを読み込む時のファイル番号
mode	モード(0=Easy 1=Normal 2=Hard)
CmdJoutai	コマンドボタンを押す時の状態
Z	コマンドボタン用
i	ループ用
a,b,c,d,a2,b2,c2	計算式作成用
Ans	計算式の答え

《第1表》プロパティ表

フォームの設定	
Name	Form1
AutoRedraw	-1 'True
BorderStyle	1 '固定(実線)
Caption	Form1
Font	ＭＳ ゴシック
…Size	9
Height	3570
KeyPreview	-1 'True
Left	0
LinkTopic	Form1
MaxButton	0 'False
ScaleHeight	213
ScaleMode	3 'ﾋﾟｸｾﾙ
ScaleWidth	297
StartUpPosition	2 '画面の中央
Top	0
Width	4545
コマンドボタンの設定	
Name	Cmd(0～2)
Caption	Command1
Font	ＭＳ Ｐゴシック
…Size	12
Height	33
Left	160
Visible	0 'False
Width	121
Name	Cmd(0)
Index	0
TabIndex	1
Top	112
Name	Cmd(1)
Index	1
TabIndex	2
Top	144
Name	Cmd(2)
Index	2
TabIndex	3
Top	176
テキストボックスの設定	
Name	TxtAns
Font	ＭＳ Ｐゴシック
…Size	24
Height	36
Left	192
MaxLength	4
TabIndex	0
Text	123
Top	72
Visible	0 'False
Width	57

CHECKER FLAG

つぐ美：前回のバージョンと大きく変更された点はないのですが，モードの追加やわたしが指摘した問題式の形などが，しっかり修正されてます。

編：もともとアイデアが良い作品でしたからね。今回の修正でゲームとしての質が向上したのではないでしょうか？

Dr.D：単純な構成ながらも操作性などがしっかり考慮されている点に注目してほしい。また，出題される問題式の形はcase文で判断しているため，若干効率が悪いようにも思えるが，その分リストからの修正や改造はやりやすいと言える。

★NECのデスクトップ・パソコンを残して，今は携帯ゲーム2個所有の状態で引きこもっている。（徳島県阿南市・T③29歳）……
【影：引きこもりとは，どういう意味でしょ。ブロードバンド時代だからこそ，コミュニケーションが大事ですよ。編：か，影さんが…。】

《リスト1》フォームのプログラム・リスト

```
Option Explicit

' コマンドボタンが押された
Private Sub Cmd_Click(Index As Integer)
  Select Case CmdJoutai
  Case "answer"
    Z = 0
    CmdJoutai = "answered"
  Case "select"
    Z = Index
  Case Else
    Exit Sub
  End Select
End Sub

' 回答入力→Enterキーでも回答できるようにする
Private Sub Form_KeyPress(KeyAscii As Integer)
  If CmdJoutai = "answer" And KeyAscii = 13 Then
    Z = 0
    CmdJoutai = "answered"
  End If
End Sub

' フォームを閉じた場合は終了する
Private Sub Form_Unload(Cancel As Integer)
  End
End Sub
```

《リスト2》モジュールのプログラム・リスト

```
Option Explicit

Declare Function GetTickCount Lib "kernel32" () As Long

Public X(3) As Integer, Ans As Integer
Public Sc As Integer, Tm As Integer, t2 As Integer, Sq As Integer, Mode As Integer
Public Seikai As Integer, Hs(2) As Integer
Public Z As Integer, CmdJoutai As String, Flag As Integer

Public Const FieldX = 300
Public Const FieldY = 240
Public Const Title = "あーる・かるきゅれーしょん Ver1.1"

' 最初に飛ぶのはここ
' エラーが出なくなるまで「' /*」と「' */」に挟まれた部分は打ち込まない方が無難です。
Sub Main()
Dim i As Integer, fnum As Integer
  ' /*
  On Error GoTo ErrHandler
  ' */
  Randomize
  ' 多重起動の防止
  If App.PrevInstance = True Then End
  ' フォームの設定
  Form1.Width = FieldX * Screen.TwipsPerPixelX
  Form1.Height = FieldY * Screen.TwipsPerPixelY
  Form1.TxtAns.Move 200, 60, 86, 44
  Form1.Caption = Title
  Form1.Show
  ' /* ハイスコアのデータを読み込む
  fnum = FreeFile
  Open MyForm & "hsc.dat" For Input As #fnum
  For i = 0 To 2
    Input #fnum, Hs(i)
  Next i
  Close #fnum
  ' */
Mark1:
  CmdJoutai = "not game"
  GameMain
  Exit Sub

  ' /* エラーが出たらここに来る
ErrHandler:
  Select Case Err
  Case 53
    For i = 0 To 2
      Hs(i) = 32 - i * 8
    Next i
    Resume Mark1
  Case Else
    MsgBox Err.Description, vbExclamation + vbOKOnly, "エラー" & Err
    End
  End Select
  ' */
End Sub

' ゲームのメイン部分
Sub GameMain()
  ' フォームが閉じられるまでずーっと繰り返される
  While 1
    GameTitle
    GameInit
    ' 時間切れになるまで繰り返される
    While (Flag = 0)
      Shutsudai
      ' 時間切れか「回答」ボタンが押されるまで繰り返される
      While (Z = -1 And Tm > 0)
        Delay 100
        t2 = t2 + 1
        If t2 >= 10 Then
          t2 = 0
          Tm = Tm - 1
          Print_Time
        End If
      Wend
      If Tm > 0 Then Hantei Else Flag = 1
    Wend
    GameOver
  Wend
End Sub

' ゲームの初期設定
Sub GameInit()
Dim i As Integer
  Form1.Cls
  Form1.Caption = Title & " [" & Mid$(" Easy Normal Hard ", Mode * 6 + 1, 6) & "]"
  Form1.TxtAns.Visible = False
  For i = 0 To 2
    Form1.Cmd(i).Visible = False
  Next i

  Sc = 0: Tm = 90: Sq = -1: t2 = 0: Flag = 0: Seikai = 0
  Msg 10, 20, "変数とその値を暗記せよ", 18, 0
  For i = 0 To 2
    X(i) = Int(Rnd * 10) + 1
    Msg 90, i * 30 + 60, "[" & Mid$("ABC", i + 1, 1) & "] = " & X(i), 24, 0
  Next i
  Delay 3000
  Form1.Cls
  Form1.Cmd(0).Caption = "回答"
  Form1.Cmd(0).Move 80, 120, 120, 40
  Form1.TxtAns.Text = ""
  CmdJoutai = "not game"
End Sub

' 問題式の作成
Sub Shutsudai()
Dim a As Integer, b As Integer, c As Integer, d As Integer, q As String
Dim a2 As String, b2 As String, c2 As String, t
  Form1.Cls
  Sq = Sq + 1: If Sq > 2 Then Sq = 0
  ' 変数sqによって項を変更
  Select Case Sq
  ' 0ならすべて数値
  Case 0
    a = Int(Rnd * 20) + 1: a2 = Mid$(Str$(a), 2)
    b = Int(Rnd * 20) + 1: b2 = Mid$(Str$(b), 2)
    c = Int(Rnd * 20) + 1: c2 = Mid$(Str$(c), 2)
```

★人が浮かれている時分に沈む人間。もうちょっとココにいてもいいですか？僕のパソコン部，ベーマガ。（愛知県名古屋市・**わいち301 27歳**）……【**影**：どうぞ，どうぞ，いつまでもここにいてかまわないですよ。**編**：その代わり影さんを追い出しますから。**影**：は？】

```
 ' 1なら1個が変数
 Case 1
   a = Int(Rnd * 3): a2 = "[" & Mid$("ABC", a + 1, 1) & "]": a = X(a)
   b = Int(Rnd * 20) + 1: b2 = Mid$(Str$(b), 2)
   c = Int(Rnd * 20) + 1: c2 = Mid$(Str$(c), 2)
   If Int(Rnd * 2) = 1 Then
     t = a: a = b: b = t
     t = a2: a2 = b2: b2 = t
   End If
 ' 2なら2個が変数
 Case 2
   a = Int(Rnd * 3): a2 = "[" & Mid$("ABC", a + 1, 1) & "]": a = X(a)
   b = Int(Rnd * 3): b2 = "[" & Mid$("ABC", b + 1, 1) & "]": b = X(b)
   c = Int(Rnd * 20) + 1: c2 = Mid$(Str$(c), 2)
 End Select
 '
 If Mode = 2 Then d = Int(Rnd * 8) + 3 Else d = Int(Rnd * 3)
 ' 変数dによって式を変更
 Select Case d
 ' 0～2はEasyレベル、Normalレベルで出てくる計算式
 Case 0: q = a2 & "+" & b2 & "=": Ans = a + b
 Case 1: q = a2 & "-" & b2 & "=": Ans = a - b
 Case 2: q = a2 & "x" & b2 & "=": Ans = a * b
 ' 3～10はHardレベルで出てくる計算式
 Case 3: q = a2 & "+" & b2 & "+" & c2 & "=": Ans = a + b + c
 Case 4: q = a2 & "+" & b2 & "-" & c2 & "=": Ans = a + b - c
 Case 5: q = a2 & "+" & b2 & "x" & c2 & "=": Ans = a + b * c
 Case 6: q = a2 & "-" & b2 & "+" & c2 & "=": Ans = a - b + c
 Case 7: q = a2 & "-" & b2 & "-" & c2 & "=": Ans = a - b - c
 Case 8: q = a2 & "-" & b2 & "x" & c2 & "=": Ans = a - b * c
 Case 9: q = a2 & "x" & b2 & "+" & c2 & "=": Ans = a * b + c
 Case 10: q = a2 & "x" & b2 & "-" & c2 & "=": Ans = a * b - c
 End Select
 Msg 140 - Len(q) * 10, 70, q, 20, 0
 Form1.TxtAns.Visible = True
 Form1.TxtAns.Text = ""
 Form1.Cmd(0).Visible = True
 Z = -1
 CmdJoutai = "answer"
 ' フォーカスがテキストボックスに来るようにする
 ' これにより、次の問題に来たときスムーズに答えが入力できるようになる
 Form1.TxtAns.SetFocus
 Print_Time
 Print_Score
End Sub

' 回答と正解のチェック
Sub Hantei()
Dim a As Integer, b As Integer
  If Val(Form1.TxtAns.Text) = Ans Then
    If Mode <> 0 Then Seikai = Seikai + 1
    ' Normal、Hardでは3問正解ごとに1変数の中身をチェンジ
    If Seikai = 3 Then
      Seikai = 0
      a = Int(Rnd * 3): b = Int(Rnd * 10) + 1
      Msg 30, 170, "正解! [" & Mid$("ABC", a + 1, 1) & "]が " & b & "に!", 18, 0
      X(a) = b
    Else
      Msg 90, 170, "正解!", 20, 0
    End If
    Sc = Sc + (1 + Sq)
    Print_Score
  Else
    Msg 20, 170, "不正解!  正解 = " & Ans, 18, 0
  End If
  Delay 2000
End Sub

' 時間切れ
Sub GameOver()
Dim s As String, i As Integer, fnum As Integer
  CmdJoutai = "not game"
  s = "そこまで! あなたの得点は " & Sc
  If Hs(Mode) < Sc Then
    Hs(Mode) = Sc
    s = s & Chr(13) & "おめでとう! 記録更新です!"
  End If
```

```
  MsgBox s, vbOKOnly, Title
  ' ハイスコアデータをファイルに書き込む
  fnum = FreeFile
  Open MyForm & "hsc.dat" For Output As #fnum
  For i = 0 To 2
    Print #fnum, Hs(i)
  Next i
  Close #fnum
End Sub

' タイトル画面
Sub GameTitle()
Dim i As Integer
  Form1.Cls
  Form1.TxtAns.Visible = False
  Form1.Caption = Title
  Msg 110, 5, "あーる", 16, 9
  Msg 50, 25, "かるきゅれーしょん", 16, 9
  Msg 80, 45, "Version 1.1", 16, 9
  Msg 60, 75, "[Easy]", 16, 0
  Msg 50, 100, "[Normal]", 16, 0
  Msg 60, 125, "[Hard]", 16, 0
  For i = 0 To 2
    Msg 150, i * 25 + 75, "High:" & Hs(i), 16, 0
  Next i
  For i = 0 To 2
    Form1.Cmd(i).Move i * 80 + 30, 150, 80, 40
    Form1.Cmd(i).Caption = Mid$(" Easy Normal Hard ", i * 6 + 1, 6)
    Form1.Cmd(i).Visible = True
  Next i
  Msg 40, 195, "(C)2001 Teikoku Soft", 14, 1
  Z = -1: CmdJoutai = "select"
  While (Z = -1): DoEvents: Wend
  Mode = Z
End Sub

' 残り時間表示
Sub Print_Time()
  Form1.Line (30, 10)-(140, 40), Form1.BackColor, BF
  Msg 20, 10, "Time:" & Format(Tm, "##0"), 20, 0
End Sub

' スコア表示
Sub Print_Score()
  Form1.Line (150, 10)-(270, 40), Form1.BackColor, BF
  Msg 150, 10, "Score:" & Format(Sc, "##0"), 20, 0
End Sub

' フォーム上に文字列を表示する命令
' Msg x,y,"メッセージ"[,s][,c]
' 座標(x,y)に大きさs(省略時12)、色c(省略時0の黒)のメッセージを表示する
Sub Msg(ByVal X As Long, ByVal y As Long, ByVal n As String, Optional ByVal s As Integer = 12,
Optional ByVal c As Integer = 0)
  Form1.CurrentX = X: Form1.CurrentY = y: Form1.FontSize = s: Form1.ForeColor = QBColor(c)
  Form1.FontBold = True: Form1.Print n
End Sub

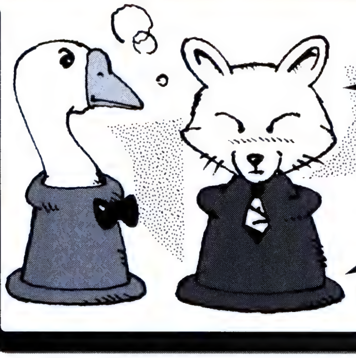
' 時間待ち命令
' Delay t
' tミリ秒待つ(t=1000で1秒)
Function Delay(ByVal t As Long)
Dim At_0 As Long, At_1 As Long
  At_0 = GetTickCount
  At_1 = At_0
  While ((At_1 - At_0) < t)
    At_1 = GetTickCount
    DoEvents
  Wend
End Function

' プログラムのあるディレクトリを取得
Function MyForm()
  If Right$(App.Path, 1) <> "¥" Then
    MyForm = App.Path & "¥"
  Else
    MyForm = App.Path
  End If
End Function
```

★うをっ，先月は原稿が間に合わなかった。パソレクが後半に。うをうをっ。（千葉県市川市・**ほおく26歳**）……【**影**：うをっ，そう言えば，パソレクの原稿まだ入っていませんね。くりひろし先生，これ見てたら大至急原稿ください。**編**：それじゃ遅いでしょ。うをっ。】

入力方法は33ページをご覧ください。

Windows(F-BASIC V6.3)用



異常に手強いCOMに勝てるか？

キツネとガチョウ

ロディマス（29歳）

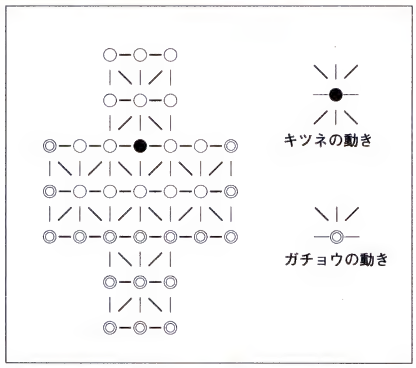
内容

西洋版，十六武蔵です。

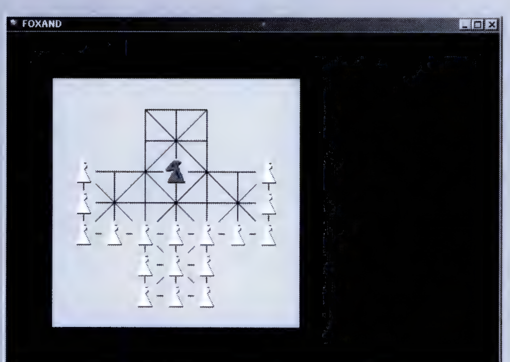
図1のように，ゲーム盤の2重丸のところにガチョウを現わす駒17個を配置します。黒い丸印のところにキツネを現わす駒を置きます。

先手はガチョウです。ガチョウは線にそってひとつ隣りの交点まで動かせます。ただし動かせるのは，左右と前，および斜め前方で，その交点があいている場所に限ります(同じ場所にふたつの駒は置けないのです)。ガチョウは駒をとびこえることができません。また後方には動かせません。ガチョウの目的はキツネの駒を追い詰めて動けなくすることです。

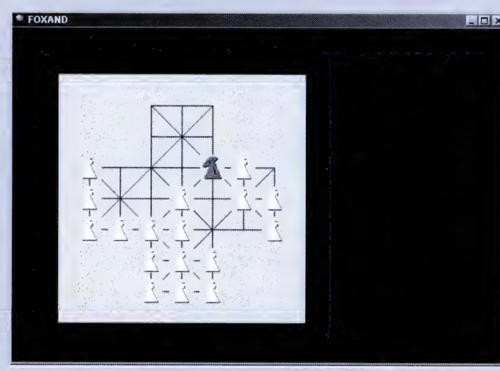
キツネの動きはもっと自由です。キツネは将棋の王将のように，前後左右斜めにひとつずつ動くことができます。チェッカーと同じように，キツネはガチョウの駒を飛び越えると，その駒をとること



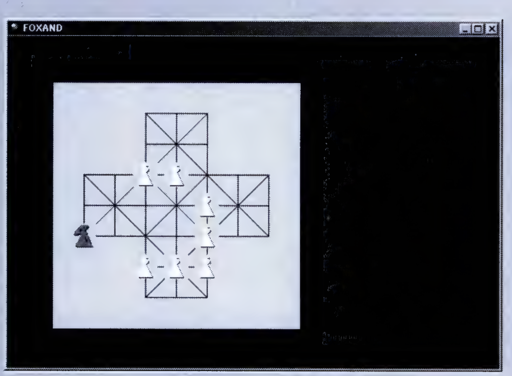
《第1図》コマの配置とコマの動き



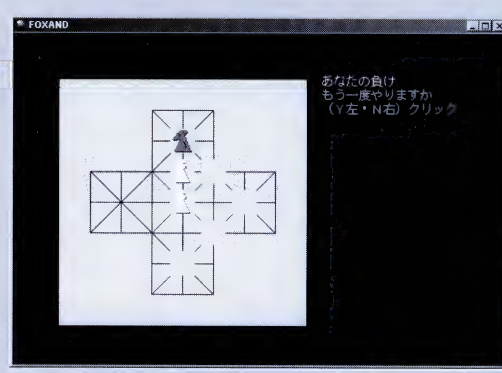
《写真1》ルールをしっかり把握してから挑戦



《写真2》COMの操作するキツネはかなり手強い



《写真3》この状態になるとさすがにキビシイ



《写真4》残念ながら負け。勝つのは至難の業

★WWGP2002が開催されると発表がありました。これを機に『ベーマガ』でもワンダーウィッチに力を入れてもらえませんか？最近，「チャレンジ・ワンダーウィッチ」のコーナーが休みで悲しいです。（東京都練馬区・畑の木29歳）……【編：松原先生，お願いします。】

《リスト》「キツネとガチョウ」のプログラム・リスト

```
'##################################################################################
'# キツネとガチョウ
#
'# 製作者 ロディマス
#
'# 使用言語 F-BASICVer6.3
#
'# OS WINDOWS98-SECOND-EDITION                                                    #
'##################################################################################
#include "windows.bi"
declare sub DISPLAY()                                   '盤と駒を表示
declare sub COMPUTER()                                          'コンピュータの思考ルーチン
declare sub MAN()                                       'プレイヤーの手番
declare sub CAPTURE_DISPLAY(OLD%,AMG%,NEW%)     '捕獲する様子を表示
declare function COM_MOVE(OLD%)                         'キツネの動きを調べる
declare function COM_CAPTURE(OLD%)                      '捕獲を調べる
declare function COM_NEW_CHECK(I%,OLD%,LST%)    '移動先のマスを調べる
declare function MAN_INPUT_CHECK(IN_OR_OUT%)    'プレイヤーの入力
declare function DIAGONAL_CHECK(OLD%)                   '斜めに移動できない場所を調べる
declare function MOUSEWAIT()                            'マウスの入力
declare function COM_CLIST_CHECK(I%,OLD%,LST%)          'コンピュータの移動をチェック
declare function MAN_MOVE_CHECK(OLD%,NEW%)      'プレイヤーの動きをチェック
var shared BOARD(16*16+2),VALUE(16*16+2) as double      '盤
var shared BLANK as object                              '
var shared BKNHT as object                              '
var shared WPAWN as object                              '
var shared FOX,GEESE,WALL,WIN_LOSE_CHECK as integer
'捕獲するアルゴリズムに使う
var shared LIST(20),NNEW(20),OOLD(20),AMNG(20) as integer
'乱数で打ち手を変える
var shared R_LST(20,20),R_NEW(20,20),R_OLD(20,20),R_AMG(20,20) as integer
var shared MAX_CAPTURE_SP(20) as integer
BITMAPOBJECT BLANK
BITMAPOBJECT BKNHT
BITMAPOBJECT WPAWN
BLANK.LOADFILE "BLANK.BMP"
BKNHT.LOADFILE "BKNHT.BMP"
WPAWN.LOADFILE "WPAWN.BMP"
FOX=2:GEESE=1:WALL=3
'盤の周囲を壁で囲む
for I%=0 to 256:BOARD(I%)=3:next I%
while 1
        cls
        '盤の中を空白にする
        for X% = 2 to 8: for Y% = 2 to 8:BOARD(X%*16+Y%) = 0:VALUE(X%*16+Y%) = 0
        next Y%: next X%
        '使わない四隅に壁を置く
        BOARD(2*16+2) =WALL:BOARD(2*16+3)=WALL:BOARD(3*16+2)=WALL:BOARD(3*16+3)=WALL
        BOARD(8*16+2) =WALL:BOARD(7*16+2)=WALL:BOARD(8*16+3)=WALL:BOARD(7*16+3)=WALL
        BOARD(2*16+8) =WALL:BOARD(2*16+7)=WALL:BOARD(3*16+8)=WALL:BOARD(3*16+7)=WALL
        BOARD(7*16+7) =WALL:BOARD(7*16+8)=WALL:BOARD(8*16+7)=WALL:BOARD(8*16+8)=WALL
        '駒を置く
        BOARD(5*16+4)=  FOX:BOARD(2*16+4)=GEESE:BOARD(2*16+5)=GEESE:BOARD(2*16+6)=GEESE
        BOARD(8*16+4) =GEESE:BOARD(8*16+5)=GEESE:BOARD(8*16+6)=GEESE:BOARD(3*16+6)=GEESE
        BOARD(4*16+6) =GEESE:BOARD(5*16+6)=GEESE:BOARD(6*16+6)=GEESE:BOARD(7*16+6)=GEESE
        BOARD(4*16+7) =GEESE:BOARD(5*16+7)=GEESE:BOARD(6*16+7)=GEESE:BOARD(4*16+8)=GEESE
        BOARD(5*16+8) =GEESE:BOARD(6*16+8)=GEESE
        'もう一つの盤上に点数を置いていく
                                VALUE(4*16+2)=-2:VALUE(5*16+2)=-1:VALUE(6*16+2)=-2
                                VALUE(4*16+3)=-1:VALUE(5*16+3)= 0:VALUE(6*16+3)=-1
        VALUE(2*16+4)=-2:VALUE(3*16+4)=-1:VALUE(4*16+4)= 0:VALUE(5*16+4)= 1:VALUE(6*16+4)=
0:VALUE(7*16+4)=-1:VALUE(8*16+4)=-2
        VALUE(2*16+5)=-1:VALUE(3*16+5)= 0:VALUE(4*16+5)= 1:VALUE(5*16+5)= 2:VALUE(6*16+5)=
```

ができます。

とった駒は盤の外へとりのけて，もう使いません。チェッカーと同じで，キツネは連続して飛ぶことができるときは，何個でもガチョウの駒をとることができます。また飛ぶことができる駒は，必ず飛ばなければならないのもチェッカーのルールと同じです。

CHECKER FLAG

つぐ美：チョット～，このCOM強すぎじゃない？あまり慣れていないゲームというのもあるけど，これに勝つのはかなり大変だわ。

Dr.D：つぐ美クンが嘆いているように，このゲームの特徴はやはりCOMルーチンじゃな。プログラムでCOMルーチンを作ること自体が，プログラム初心者にとって難しい部類になるのは，ベーマガ読者ならわかると思うが，この作品はそのCOMルーチンを強く作り上げているわけじゃ。そういう意味でもこの作品のレベルの高さが感じられるのではないかな？リストの解析は難しいと思うが，こういったプログラムのリストに一度は目を通しておくのも勉強になるじゃろう。

キツネはガチョウの包囲網をくぐりぬけて，ガチョウの数を残り2個にすれば勝ちです。

このゲームのいいところは引き分けがないことです。

操作・実行説明

コンピュータがキツネを受け持ちます。あなたはガチョウをマウスの左クリックで動かし，キツネを動けなくしてください。コンピュータは強いですぞ！

実行には画像ファイルが必要になります。BKNHT.BMP，WPAWN.BMP，BLANK.BMPの三つの画像ファイルを用意してください。BKNHT.BMP，WPAWN.BMPはコマの画像，BLANK.BMPは背景色のみで構成される画像となります。サイズはすべて32×32です。

苦労した点

キツネの連続ジャンプによる捕獲が難しかったです。ただ連続ジャンプしていくのではなく，一番多くガチョウを獲れるジャンプをするようにしてあります(そのかわりプログラムが長くなってしまいました)。

その分コンピュータはかなり強く，作った本人も5回に1回くらいしか勝てません。このルーチンを利用して，今度はチェッカーに挑戦してみようと思います。

★参考にしたもの

・「世界のゲーム事典」松田道弘著　東京堂出版

画像ファイル

BLANK.BMPの画像



WPAWN.BMPの画像



BKNHT.BPMの画像

★最近，多くの友人からパソコンのサポート依頼が殺到。なかには某メーカー製パソコンの改造まで…。忙しい毎日です。（東京都江戸川区・やん23歳）……【編：社内では，ベーマガ編集部が各部署から引っ張りだこです。PCがフリーズしたとか…。】

```
1:VALUE(7*16+5)=-0:VALUE(8*16+5)=-1
          VALUE(2*16+6)=-2:VALUE(3*16+6)=-1:VALUE(4*16+6)= 0:VALUE(5*16+6)= 1:VALUE(6*16+6)=
0:VALUE(7*16+6)=-1:VALUE(8*16+6)=-2
                              VALUE(4*16+7)=-1:VALUE(5*16+7)= 0:VALUE(6*16+7)=-1
                              VALUE(4*16+8)=-2:VALUE(5*16+8)=-1:VALUE(6*16+8)=-2

          DISPLAY:WIN_LOSE_CHECK=0
          while 1
                    MAN:DISPLAY:COMPUTER
                    if WIN_LOSE_CHECK<>0 then goto *ENDING
                    wait 10:DISPLAY
          wend
          *ENDING
          if WIN_LOSE_CHECK=1 then locate 50,3:print "私の負けです"
          if WIN_LOSE_CHECK=2 then locate 50,3:print "あなたの負け"
          locate 50,4:print "もう一度やりますか"
          locate 50,5:print "（Y左・N右）クリック"
          if MOUSEWAIT=1 then goto *BREAK
wend
*BREAK: BLANK.DELETEOBJECT:BKNHT.DELETEOBJECT:WPAWN.DELETEOBJECT:end

'盤と駒を表示
sub DISPLAY()
var I%,J%
          line(40*1+20,40*1+20)-(40*9+20,40*9+20),pset,9,bf
          for I%=2 to 3:line(40* 4+20,40*I%+20)-(40* 6+20,40*I%+20),pset,0:next I%
          for I%=7 to 8:line(40* 4+20,40*I%+20)-(40* 6+20,40*I%+20),pset,0:next I%
          for I%=2 to 3:line(40*I%+20,40* 4+20)-(40*I%+20,40* 6+20),pset,0:next I%
          for I%=7 to 8:line(40*I%+20,40* 4+20)-(40*I%+20,40* 6+20),pset,0:next I%
          for I%=4 to 6:line(40* 2+20,40*I%+20)-(40* 8+20,40*I%+20),pset,0
                         line(40*I%+20,40* 2+20)-(40*I%+20,40*8 +20),pset,0:next I%
          line(40*4+20,40*2+20)-(40*8+20,40*6+20),pset,0
          line(40*4+20,40*4+20)-(40*6+20,40*6+20),pset,0
          line(40*2+20,40*4+20)-(40*6+20,40*8+20),pset,0
          line(40*2+20,40*6+20)-(40*6+20,40*2+20),pset,0
          line(40*4+20,40*6+20)-(40*6+20,40*4+20),pset,0
          line(40*4+20,40*8+20)-(40*8+20,40*4+20),pset,0
          for I%=2 to 8:for J%=2 to 8
                    if BOARD(I%*16+J%)= FOX   then DRAWBITMAP BKNHT,40*I%+4,40*J%+4
                    if BOARD(I%*16+J%)= GEESE then DRAWBITMAP WPAWN,40*I%+4,40*J%+4
          next J%:next I%
end sub

'斜めに動けるかチェック
function DIAGONAL_CHECK(MEASURE%)
          DIAGONAL_CHECK=0
          if MEASURE%= 37 or MEASURE%= 52 then DIAGONAL_CHECK=1
          if MEASURE%= 54 or MEASURE%= 67 then DIAGONAL_CHECK=1
          if MEASURE%= 69 or MEASURE%= 71 then DIAGONAL_CHECK=1
          if MEASURE%= 82 or MEASURE%= 84 then DIAGONAL_CHECK=1
          if MEASURE%= 86 or MEASURE%= 88 then DIAGONAL_CHECK=1
          if MEASURE%= 99 or MEASURE%=101 then DIAGONAL_CHECK=1
          if MEASURE%=103 or MEASURE%=116 then DIAGONAL_CHECK=1
          if MEASURE%=118 or MEASURE%=133 then DIAGONAL_CHECK=1
end function

'捕獲していく様子を表示
sub CAPTURE_DISPLAY(OLD%,amg%,NEW%)
var I%,J%
          I%=(NEW% ¥ 16):J%=(NEW% mod 16):DRAWBITMAP BKNHT,40*I%+4,40*J%+4
          I%=(AMG% ¥ 16):J%=(AMG% mod 16):DRAWBITMAP BLANK,40*I%+4,40*J%+4
          I%=(OLD% ¥ 16):J%=(OLD% mod 16):DRAWBITMAP BLANK,40*I%+4,40*J%+4:wait 50
end sub

sub COMPUTER()
var OLD%,X%,Y%,SP%,I%,J%,MOVE_CHECK%,amg%,CAPTURE_SP%,MOUSECHECK%
          EV2%=-30000:MOVE_CHECK%=0:CAPTURE_SP%=0:AMG%=0:SP%=0
          '狐の位置を探し出しＯＬＤ%に保存する
          for I%=0 to 256
                    if BOARD(I%)=2 then OLD%=I%
          next I%
          '捕獲できなかったら普通のムーブをする
          MOVE_CHECK%=COM_CAPTURE(OLD%)
          if MOVE_CHECK%=0 then MOVE_CHECK%=COM_MOVE(OLD%)
          '狐が動けなくなったらコンピュータの負け
          if move_check%=0 then WIN_LOSE_CHECK=1
          'ガチョウの数が２個になったらプレイヤーの負け
          GEESE%=0:for X%=2 to 8:for Y%=2 to 8
                    if BOARD(X%*16+Y%)=1 then GEESE%=GEESE%+1
          next Y%:next X%
          if GEESE%<=2 then WIN_LOSE_CHECK=2
end sub

'普通のムーブ
function COM_MOVE(OLD%)
var lst%,SP%,EV1%,EV2%,CHECK%
          EV2%=-3000:CHECK%=0:lst%=0:SP%=0
          while 1
                    EV1%=0:LST%=COM_CLIST_CHECK(0,OLD%,LST%)
                    if CHECK%=0 and LST%>10 then goto *RET
                    if LST%>10 then goto *BREAK
                    I%=COM_NEW_CHECK(0,OLD%,LST%):EV1%=VALUE(I%):LST%=LST%+1
                    if EV1%<EV2% then goto *JUMP
                    if EV1%>EV2% then SP%=0:NNEW%(SP%)=I%:EV2%=EV1%:CHECK%=1:goto *JUMP
                    SP%=SP%+1:NNEW%(SP%)=I%
                    *JUMP
          wend
          *BREAK:SP%=cint(rnd(1)*SP%):NEW%=NNEW%(SP%):BOARD(OLD%)=0:BOARD(NEW%)=2:beep
          I%=(NEW%  ¥ 16):J%=(NEW%  mod 16):DRAWBITMAP BKNHT,40*I%+4,40*J%+4
          I%=(OLD%  ¥ 16):J%=(OLD%  mod 16):DRAWBITMAP BLANK,40*I%+4,40*J%+4
*RET:     COM_MOVE=CHECK%
end function

function COM_NEW_CHECK(I%,OLD%,lst%)
var J%
          if lst%=0 then j%=+16
          if lst%=1 then j%=-16
          if lst%=2 then j%=+1
          if lst%=3 then j%=-1
          if lst%=4 then j%=+15
          if lst%=5 then j%=-15
          if lst%=6 then j%=+17
          if lst%=7 then J%=-17
          if I%=1 then J%=J%*2
          COM_NEW_CHECK=OLD%+J%
end function

function COM_CLIST_CHECK(I%,OLD%,lst%)
var CHECK%
          CHECK%=0
          if lst%=0 then goto *LIST0
          if lst%=1 then goto *LIST1
          if lst%=2 then goto *LIST2
          if lst%=3 then goto *LIST3
          if lst%=4 then goto *LIST4
          if lst%=5 then goto *LIST5
          if lst%=6 then goto *LIST6
          if lst%=7 then goto *LIST7
          lst%=100:goto *ret
```

★最近パソコン遅いです，助けて影さん！（栃木県河内郡・DASH14歳）……【**影**：まずは，アプリケーション・ソフトをすべて消して…。初期のスタイルにもどしましょう。**つぐ美**：ちょっと，それじゃ，なんの解決にもなってないでしょ。**影**：まずはメモリ増設を。】

```
*LIST0:    if I%=1 and BOARD(OLD%+16)=1 and BOARD(OLD%+32)=0 then lst%=0:goto *ret
           if I%=0 and BOARD(OLD%+16)=0                      then lst%=0:goto *ret
*LIST1:    if I%=1 and BOARD(OLD%-16)=1 and BOARD(OLD%-32)=0 then lst%=1:goto *ret
           if I%=0 and BOARD(OLD%-16)=0                      then lst%=1:goto *ret
*LIST2:    if I%=1 and BOARD(OLD%+ 1)=1 and BOARD(OLD%+ 2)=0 then lst%=2:goto *ret
           if I%=0 and BOARD(OLD%+ 1)=0                      then lst%=2:goto *ret
*LIST3:    if I%=1 and BOARD(OLD%- 1)=1 and BOARD(OLD%- 2)=0 then lst%=3:goto *ret
           if I%=0 and BOARD(OLD%- 1)=0                      then lst%=3:goto *ret
*LIST4:    '或る場所にある駒は斜めに動けません
           CHECK%=DIAGONAL_CHECK(OLD%):if CHECK%=1 then LST%=100:goto *ret
           if I%=1 and BOARD(OLD%+15)=1 and BOARD(OLD%+30)=0 then lst%=4:goto *ret
           if I%=0 and BOARD(OLD%+15)=0                      then lst%=4:goto *ret
*LIST5:    if I%=1 and BOARD(OLD%-15)=1 and BOARD(OLD%-30)=0 then lst%=5:goto *ret
           if I%=0 and BOARD(OLD%-15)=0                      then lst%=5:goto *ret
*LIST6:    if I%=1 and BOARD(OLD%+17)=1 and BOARD(OLD%+34)=0 then lst%=6:goto *ret
           if I%=0 and BOARD(OLD%+17)=0                      then lst%=6:goto *ret
*LIST7:    if I%=1 and BOARD(OLD%-17)=1 and BOARD(OLD%-34)=0 then lst%=7:goto *ret
           if I%=0 and BOARD(OLD%-17)=0                      then lst%=7:goto *ret
           lst%=100
*RET:      COM_CLIST_CHECK=LST%
end function

sub MAN()
var X%,Y%,OLD%,NEW%,CHECK%,MOVE_CHECK%,KOMA%
           MOVE_CHECK%=0
           while MOVE_CHECK%=0
                      OLD%=MAN_INPUT_CHECK(1):KOMA%=BOARD(OLD%):NEW%=MAN_INPUT_CHECK(2)
                      MOVE_CHECK%=MAN_MOVE_CHECK(OLD%,NEW%)
                      if MOVE_CHECK%=0 then locate 10,1:print "そこには進めません。":X%=(OLD% ¥
16):Y%=(OLD% mod 16):line (X%*40+3,Y%*40+3)-step(32,32),pset,9,b
           wend:BOARD(OLD%)=0:BOARD(NEW%)=KOMA%
end sub

function MAN_INPUT_CHECK(IN_OR_OUT%)
var CHECK%,CELL%
           while CHECK%=0
                      MOUSECHECK=MOUSEWAIT
                      X%=int(mouse(0)/40):Y%=int(mouse(1)/40)
                      CELL%=X%*16+Y%:locate 10,1:print "                    "
                      if CELL%<0 and CELL%>255 then goto *JUMP
                      if IN_OR_OUT%=2 then CHECK%=1:goto *JUMP
                      if BOARD(CELL%)>0 and BOARD(CELL%)<17 then CHECK%=1:line (X%*40+3,Y%*40+3)-
step(32,32),pset,5,b else locate 10,1:print "其処に駒はありません"
                      *JUMP
           wend:MAN_INPUT_CHECK=CELL%
end function

function MAN_MOVE_CHECK(OLD%,NEW%)
var CHECK%
           MAN_MOVE_CHECK=0
           if BOARD(NEW%)<>0 then goto *RET
           'ガチョウは後ろ向きには進めません
           if NEW%=OLD%+16 then MAN_MOVE_CHECK=1:'左
           if NEW%=OLD%-16 then MAN_MOVE_CHECK=1:'右
           if NEW%=OLD%- 1 then MAN_MOVE_CHECK=1:'前
           '或る場所にある駒は斜めに動けません
           CHECK%=DIAGONAL_CHECK(OLD%)
           if CHECK%=1 then goto *RET
           if NEW%=OLD%+15 then MAN_MOVE_CHECK=1:'左前
           if NEW%=OLD%-17 then MAN_MOVE_CHECK=1:'右前
*RET
end function

'右クリックの時１を左クリックの時０を還す
function MOUSEWAIT()
           while mouse(2,1)<>-1 and mouse(2,0)<>-1
                      if mouse(2,1)=-1 then MOUSEWAIT=1 else  MOUSEWAIT=0
           wend:while mouse(2,1)=-1 or mouse(2,0)=-1:wend
end function

'##################################################################################
'# 連続キャプチャーを調べるアルゴリズムです。この部分が一番苦労しました。色々コメントを   #
'# 付けていますが、自分でも処理の流れを追っていて頭が痛くなってきます。                   #
'##################################################################################
function COM_CAPTURE(OLD%)
var X%,Y%,SP%,I%,J%,MOVE_CHECK%,AMG%,CAPTURE_SP%
           '変数を初期化
           EV2%=-30000:MOVE_CHECK%=0:CAPTURE_SP%=0:AMG%=0:SP%=0:MOVE_CHECK%=0
           *LOOP1:    LST%=COM_CLIST_CHECK(1,OLD%,0)
                      '捕獲出来ない時は普通の移動をする
                      if MOVE_CHECK%=0 and LST%>10 then goto *RET
                      '捕獲出来る時はチェックをする
                      if MOVE_CHECK%=0 and LST%<10 then MOVE_CHECK%=2:goto *JUMP2
                      '連続捕獲が終わったらループから抜ける
                      if MOVE_CHECK%=2 and LST%>10 then goto *BREAK
           *JUMP2:    NEW%=COM_NEW_CHECK(1,OLD%,LST%):AMG%=int(OLD%+NEW%)/2

LIST(CAPTURE_SP%)=LST%:NNEW(CAPTURE_SP%)=NEW%:OOLD(CAPTURE_SP%)=OLD%:AMNG(CAPTURE_SP%)=AMG%:CAPTURE_SP%=CA
PTURE_SP%+1
                      BOARD(OLD%)=0:BOARD(AMG%)=0:BOARD(NEW%)=2
                      OLD%=NEW%
           goto *LOOP1
*BREAK:    CAPTURE_SP%=CAPTURE_SP%-1
           if CAPTURE_SP%<0 then goto *break2
           '捕獲する手筋を記録していく
           EV1%=VALUE(NEW%)+CAPTURE_SP%
           if EV1%<EV2% then goto *LOOP2
           if EV1%>EV2% then SP%=0:EV2%=EV1%:goto *JUMP11
           SP%=SP%+1
           *JUMP11
           MAX_CAPTURE_SP(SP%)=CAPTURE_SP%
           for I%=0 to CAPTURE_SP%
                      R_LST(I%,SP%)=LIST(I%):R_NEW(I%,SP%)=NNEW(I%)
                      R_OLD(I%,SP%)=OOLD(I%):R_AMG(I%,SP%)=AMNG(I%)
           next I%
           '手を戻していくルーチン
           *LOOP2:    LST%=LIST(CAPTURE_SP%):NEW%=NNEW(CAPTURE_SP%)
                      OLD%=OOLD(CAPTURE_SP%):AMG%=AMNG(CAPTURE_SP%)
                      '駒を一手戻す
                      BOARD(NEW%)=0:BOARD(OLD%)=2:BOARD(AMG%)=1
                      '別の手を探す
                      LST%=LST%+1:LST%=COM_CLIST_CHECK(1,OLD%,LST%)
                      if LST%<10 then goto *JUMP2
                      '別の手が無いときはさらに一手戻す
                      CAPTURE_SP%=CAPTURE_SP%-1
                      if CAPTURE_SP%<0 then goto *break2
           goto *LOOP2
*BREAK2:if MOVE_CHECK%<>2 then goto *RET
           SP%=cint(rnd(1)*SP%)
           CAPTURE_SP%=MAX_CAPTURE_SP(SP%)
           for I%=0 to CAPTURE_SP%
                      LST%=R_LST(I%,SP%):NEW%=R_NEW(I%,SP%)
                      OLD%=R_OLD(I%,SP%):AMG%=R_AMG(I%,SP%)
                      BOARD(NEW%)=2:BOARD(OLD%)=0:BOARD(AMG%)=0
                      beep:CAPTURE_DISPLAY OLD%,AMG%,NEW%
           next I%
*RET:      COM_CAPTURE=MOVE_CHECK%
end function
```

★愛読者カードのキリトリ線を手で切れるようにしてほしい。（東京都北区大路・Ike16歳）……**【影：私は手で切れるぞ。ホラ。つぐ美：切り跡がきたな〜い！影：**ということなので，ハサミやカッターを使いましょう。**つぐ美：**うーん，なんか話が続かない。**】**

Windows(HSP V2.5)用



2匹のサルを落とさないように

サル(さる)も木(き)から落(お)ちないで

三好　利治（46歳）

入力方法は33ページをご覧ください。

内容

サルが木から落ちないうちに，たくさんの語句をタイプしていくという，タッチタイピング練習ソフトです。

また出題語句は，自分が練習したい日本語，その英語スペル、読みとしてのローマ字を作って，このプログラムから出題させることが出来るようにしました。

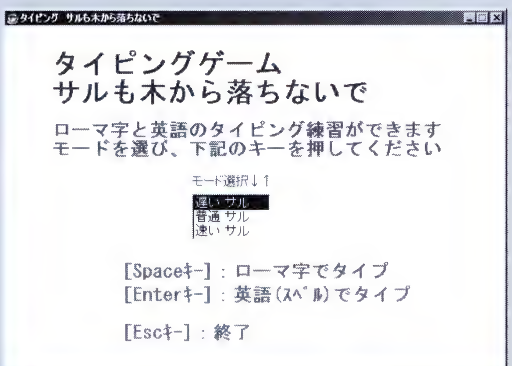
使用方法

起動してみればすぐわかります。マウスは使いません。終了はEscキーです。

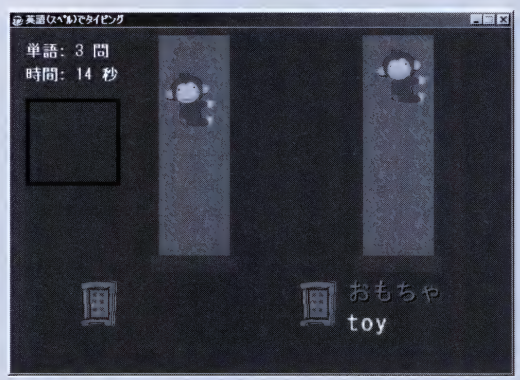
初期画面で，サルのモードを選択（**写真1**）した後，EnterキーまたはSpaceキーで英語スペルかローマ字かを選択すると，タイプゲーム画面に進みます（**写真2**）。

サルが地面まで落ちるとゲームオーバーです。単語クリアごとにサルが上に戻ります。左右のサルは独立して動いていますので，スペースキーでうまく左右を選び（**写真3**），その単語をクリアして出来るだけ多くの単語を入力しましょう。

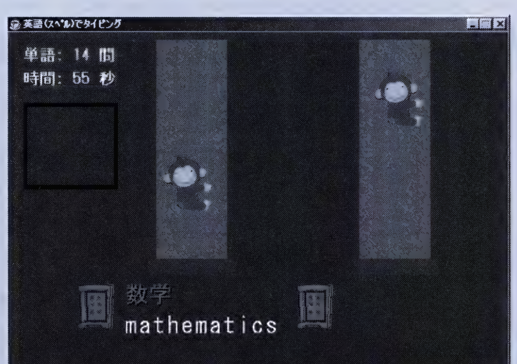
ゲームオーバーになるとランキング，



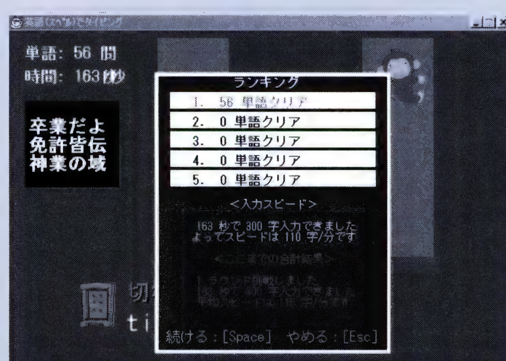
《写真１》お好みの設定にしてゲームスタート



《《写真２》表示される単語をどんどんタイピング



《写真３》左右のサルの位置を気にしながら進めていく



《写真４》ゲームオーバー後にタイピング情報を表示

★「人もハードディスクが付けれればね」「知識だけじゃだめさ。応用力がなければ」「ウフフ…そうね？」（愛知県名古屋市・ズオウ・リアキ29歳）……【影：いったい誰と誰が会話してるんでしょう？そこが気になる！編：悩んでないで，原稿早く仕上げてくださいね。】

《リスト1》「サルも木から落ちないで」のプログラム・リスト

```
;==================================
;   英語スペル・ローマ字タイプ練習
;   タイピングゲーム  サルも木から落ちないで
;   program name ERtypsaru    2001/10 tosiharu.miyoshi  hsp2.5
;              v1.0→        2001/10
;==================================
;
;配列、変数など
        sdim eigbuf,1,26              ;英文字表示比較用
        eigbuf="a","b","c","d","e","f","g","h","i","j","k","l","m","n"
        eigbuf.14="o","p","q","r","s","t","u","v","w","x","y","z"
        sdim jap,24,300               ;日本語データ格納用max300
        sdim eig,24,300               ;英語データ  ""
        sdim rom,24,300               ;ローマ字データ ""
        dim rank,6                    ;ランキング用入力単語数格納用
        dim rakkax,201                ;落下スピード用0
        dim rakkay,201                ;落下スピード用1
        dim rakkaz,201                ;落下スピード用2
;                                    ;単語クリア数によるスピード量(確率)
        rakkax.0=30:rakkay.0=20:rakkaz.0=15  ;(自分用に変えましょう)
        repeat 10
          a=cnt+1:b=cnt+11:c=cnt+21:d=cnt+31
          rakkax.a=30:rakkax.b=20:rakkax.c=15:rakkax.d=8
          rakkay.a=20:rakkay.b=15:rakkay.c=10:rakkay.d=5
          rakkaz.a=15:rakkaz.b=10:rakkaz.c=5 :rakkaz.d=3
        loop
        repeat 60
          a=cnt+41
          rakkax.a=5:rakkay.a=3:rakkaz.a=2
        loop
        repeat 100
          a=cnt+101
          rakkax.a=2:rakkay.a=2:rakkaz.a=2
        loop
;
        mmm=0                 ;データファイルのmaxデータ 乱数生成数に使用
        tengou=0:secgou=0:mojgou=0:kaigou=0  ;入力文字数、秒数など
        randomize
;
;画像,wavファイルなどのload
        buffer 5,,,0:picload "tosi.bmp"  ;訂正イベント画像
      buffer 6,,,0:picload "saru.bmp"  ;サルの画像
        sndload "pon.wav",2
        sndload "kan.wav",3:sndload "tigau.wav",4
;
;データファイルをロード 配列へセーブ
        gosub *tread
;
;メインルーチン
*thajime
        screen 0,640,440,0,dispx-640/2,dispy-440/2
        title "タイピング  サルも木から落ちないで"
        color 248,248,200:boxf
        color 0,0,255:font "MS ゴシック",36,1
        pos 60,30:mes "タイピングゲーム"
        mes "サルも木から落ちないで"
        color 0,128,0:font "MS ゴシック",24,1
        pos 60,120
        mes "ローマ字と英語のタイピング練習ができます"
        mes "モードを選び、下記のキーを押してください"
        color 128,128,0
```

スピードが表示されます(**写真4**)。

改造その他

問題は,japanese.txtから表示，答えがroman.txtまたはenglish.txtから表示というわけです。よって別に英語，ローマ字でなくても自分で好きに変えて使うことも出来ます。

repeat-loopの命令においてネストの問題があって(うまく説明できませんが)ループはわかりづらくなってしまいました。他の言語のようなforやdoならネストの深さは気にしなくてもいいのでしょうか？

まず，キータッチの個人差や各自PCのキーボード設定(リピートの速さなど)で，プログラムの反応に多少違和感が生じる場合がありますので，ソースに書いてあるようにawaitを調整してください。HSPでは押した瞬間だけを取得する命令がないものですから(英字入力の場合)。モジュールを作ると，今度は感度が鈍く，速く打てなかったのです。何かいい方法があれば，教えて頂くとありがたいと思います。

サルモードでの落ちる速さや，単語数に応じた加速を変えたりして難易度を調整するとか，カッコいい背景画像，落ちる画像を変えても面白いと思います。また，落ちるのを横にして，なにかに追いかけられて捕まったらオーバーというのもいいかも…。

もし他に興味がおありなら，私の趣味サイトへ一度どうぞ(チャウチャウとイラストがメインだったりしますが)。

http://www.h3.dion.ne.jp/~yukomaru/index.htm

☆　☆　☆　☆　☆

CHECKER FLAG

つぐ美：基本的なタイピングゲームですが，2匹のサルを使ったゲーム設定やスペル・ローマ字切り替えができるなど，いろんなところで独自のアイデアが盛り込まれている作品です。

編：問題文の入力がチョット面倒そうですが，テキストファイルなので自分で好きな問題に書き換えられますね。

Dr.D：気になったのが，機種によって生じるキー入力の反応の遅さじゃな。キー入力をメインとするタイピングゲームで，この不具合があるのはチト残念じゃな。説明文でも記述してあるように，いろいろと工夫したみたいじゃが，やはりソフト側でウェイトを自動調整してくれるなどの工夫はほしかったところじゃ。



saru.bmpの画像



tosi.bmpの画像

★ベーマガ読者にはH"ユーザーが多いような気が…。僕もfeel H"ですよ。フリップ型の新機種がほしい。(群馬県前橋市・**トランス19歳**)
……【**つぐ美**：読者のみなさんは，携帯電話やPHSは，電話機能としてよりもメールとして使われているほうが多いのかなぁ。】

```
    pos 150,300:mes "[Spaceキー]: ローマ字でタイプ"
    pos 150,330:mes "[Enterキー]: 英語(スペル)でタイプ"
    mes " "
    mes "[Escキー]: 終了"
;
    font "MSゴシック",16:color 64,64,64
    pos 240,190:mes "モード選択↓↑"
    objsize 100,25
    tck=0               ;モード選択変数デフォルトID=0
    pos 240,215
    listbox tck,50,"遅い サル¥n普通 サル¥n速い サル" :ID=0,1,2
;
    tdata=mmm           ;単語量(maxデータ)乱数生成数
    gko=0:gtm=0:gtan=0  ;文字計、時間計、単語数
    syy1=10:syy2=10     ;サルの最初のy位置
;
*tskey
    wait 1:stick key:if key=0:goto *tskey ;選択
    if key=16:sw=2:goto *thajimeru        ;ローマ字
    if key=32:sw=1:goto *thajimeru        ;英語
    if key=128:goto *tend                 ;end
    goto *tskey
;
*thajimeru
    cls
    color 0,0,255:boxf
    if sw=1:title "英語(スペル)でタイピング"
    if sw=2:title "ローマ字でタイピング"
    sayuu=1  ;右か左かデフォルトは左
    gtan=0   ;入力単語数
;
    color 0,0,255:boxf 20,20,120,100
    font "MS ゴシック",20,1:color 255,255,255
    pos 20,20:mes "単語: "+gtan+" 問"
    pos 20,50:mes "時間: "+gtm+" 秒"
    color 0,0,0:boxf 20,90,140,200        ;メッセージ用
    color 0,0,255:boxf 25,95,135,195      ;   ""
    color 128,128,0:boxf 190,10,280,290 ;サルの通路
    color 128,64,0:boxf 180,290,290,310 ;地面
    color 128,128,0:boxf 450,10,540,290 ;サルの通路右
    color 128,64,0:boxf 440,290,550,310 ;地面右
    gmode 2
    pos 200,syy1:gcopy 6,0,0,60,68   ;サルの開始表示
    pos 470,syy2:gcopy 6,0,0,60,68   ;サルの開始表示右
;
    font "MS ゴシック",20,1
    color 228,228,228:boxf 120,205,500,260
    color 0,128,0
    pos 140,210:mes "スペースを押すとスタートします"
    color 128,0,0
    pos 145,235:mes "左右移動はスペースを押します"
;
*tskey2
    wait 1:stick key:if key=0:goto *tskey2
    if key=16:goto *tgame     ;スペースキー押下ゲームスタートへ
    goto *tskey2
;
*tgame
    wait 10
    color 0,0,255:boxf 120,205,500,260  ;スタート文字を消す
    color 128,128,0:boxf 190,10,280,290 ;サルの通路
    color 128,64,0:boxf 180,290,290,310 ;地面
    color 128,128,0:boxf 450,10,540,290 ;サルの通路右
```

```
    color 128,64,0:boxf 440,290,550,310 ;地面右
    pos 200,syy1:gcopy 6,0,0,60,68      ;サルの開始表示
    pos 470,syy2:gcopy 6,0,0,60,68      ;サルの開始表示右
    color 0,0,255:boxf 100,320,640,440
    dame=0  ;ゲームオーバー判定
;
;ここからタイプ入力繰返しjj回
    jj=-1
;
*tgamekuri
    wait 1                          ;ゲームオーバーまでループ
    jj=jj+1   ;0から開始 repeat-loopはネストの問題で?使わず
    pos 80,320:gcopy 5,128,0,64,64  ;訂正イベント画像
    pos 360,320:gcopy 5,128,0,64,64 ;   ""   右
    redraw 1
*thyouji
    rnd t,tdata:tjap=jap.t                    ;MAX tdata個のデータ
  if sw=1:teig=eig.t:else:teig=rom.t  ;英語かローマ字か
    font "MS ゴシック",30
    if sayuu=1{                      ;左
       color 0,0,255:boxf 430,320,640,440 ;右を消す
       color 0,0,0:pos 152,322
       mes tjap                      ;問題表示黒
       color 255,0,255:pos 150,320
       mes tjap                      ;問題表示ピンク
    }
    else{                            ;右
       color 0,0,255:boxf 150,320,360,440 ;左を消す
       color 0,0,0:pos 432,322
       mes tjap                      ;問題表示黒
       color 255,0,255:pos 430,320
       mes tjap                      ;問題表示ピンク
    }
;
*tkey
    color 255,255,255
    strlen nagasa,teig
    sdim keybuf,1,24
    buf=""
  font "MS ゴシック",30,0
;
    repeat nagasa
      strmid buf,teig,cnt,1
      keybuf.cnt=buf
      if sayuu=1:pos cnt*18+150,360:else:pos cnt*18+430,360
      mes buf
    loop
;
    gosub *tgettime1          ;1単語開始,時間計算1へ
;
    repeat nagasa             ;1単語nagasa文字分の入力
*tttt
       nest=0                 ;ネスト深防止用フラグ
       await 1
       stick key
       if key=16{             ;スペースキーで左右変更
          if sayuu=1:sayuu=2:else:sayuu=1
          nest=1
          break     ;ループを抜けてから再問題表示へ(ネストの問題)
       }
       key=0:jjj=-1
*tloop2                     ;英字キーの判定 repeatloopは使わない
       jjj=jjj+1:if jjj>90:goto *tnopush  ;入力なし
```

```
        getkey key,jjj+65                  ;英字キー(a-z)
        if key=1:k=jjj:goto *tkeypush      ;押した,k番目の英字取得
        goto *tloop2
;
*tnopush                        ;入力なし待ち状態の処理
                                  ;クリア数に応じ落ちるスピード(確率)代入
        if tck=0:rsuu=rakkax.gtan   ;遅いサル
        if tck=1:rsuu=rakkay.gtan   ;普通サル
        if tck=2:rsuu=rakkaz.gtan   ;速いサル
        rnd ss,rsuu:if ss=0:gosub *tsaru ;落ちる処理へ
        if dame=1:goto *tkekka   ;ゲームオーバー結果へ
        goto *tttt               ;キー入力待ちへ
;
*tkeypush                       ;英字が入力された
        color 0,0,255              ;訂正の絵を消す
        if sayuu=1:boxf 16,330,80,440:else:boxf 296,330,360,440
;
                                    ;英字バッファeigbufは 0から25
      if keybuf.cnt=eigbuf.k{    ;1文字タイプ正解,青色で表示
          color 0,0,128
          if sayuu=1:pos cnt*18+150,360:else:pos cnt*18+430,360
          mes keybuf.cnt
          await 185          ;各自のキーボード設定により変えて下さい
        }
        else{                      ;1文字タイプミス,赤色で表示
          gosub *ttigau              ;ミスだよイベントへ
          color 255,0,0
          if sayuu=1:pos cnt*18+150,360:else:pos cnt*18+430,360
          mes keybuf.cnt
          await 185          ;各自のキーボード設定により変えて下さい
          goto *tttt                 ;今の1文字やり直し
        }
      loop
      if nest=1:goto *thyouji   ;スペース押下した再度問題表示へ
;
*ttanend                             ;1単語終了
      gosub *tgettime2               ;時間計算2へ(time2-time1で計算)
      if sayuu=1:gosub *tagaru1      ;左サルが元の位置に戻る
      if sayuu=2:gosub *tagaru2      ;右サルが元の位置に戻る
      snd 2                          ;ポン効果音
;
      strlen ko,teig                 ;一単語の文字の数(長さ)
      gko=gko+ko:mojgou=mojgou+ko    ;入力した文字数計
;
      color 0,0,255:boxf 20,20,120,85  ;回数、秒数を消す
      font "MS ゴシック",20,1:color 255,255,255
;
      gtan=jj+1                              ;入力出来た単語数
      pos 20,20:mes "単語: "+gtan+" 問"
      pos 20,50:mes "時間: "+gtm+" 秒"
      redraw 0
    color 0,0,255
      boxf 16,320,360,440:boxf 296,320,640,440
;
    goto *tgamekuri  ;終わる(落ちる)まで繰り返す
;
;結果表示(ランキングとタイプスピードなど)
;別ウィンドウに
*tkekka
      snd 3:wait 20
      gosub *thyouka   ;評価イベントへ
      wait 200
;                     ;今回単語数0なら評価せず初期画面に戻る
```

```
        if gtan=0:goto *thajime
;
        bgscr 2,300,360,1,dispx-300/2,dispy-360/2 ;評価画面
        gsel 2,1
        color 255,255,255:boxf
        color 0,0,128:boxf 5,5,295,355
;
        kaigou=kaigou+1      ;最後集計用
;
        color 255,255,255
        fun=gko*60/gtm                     ;今回スピード(小数誤差補正)
        fungou=mojgou*60/secgou            ;合計の平均スピード
;
        rank.5=gtan  ;今回の単語数をとりあえず最後に格納
;
;rank.0~rank.5をソートする;データ個数id=6 バブルソート
;
        id=6
        repeat id-1
          nct=cnt
            repeat id-nct-1
              j=cnt
              ne=j+1
              if (rank.j >= rank.ne):goto *tno
        ;スワップ
              tmp=rank.j
              rank.j=rank.ne:rank.ne=tmp
*tno
            loop
        loop
;
        repeat 5                              ;5位以内か
          if gtan=rank.cnt:jyun=cnt:break
        loop
;
        font "MS ゴシック",16,1:color 255,255,255
        pos 100,5:mes "ランキング"
        boxf 20,25,280,45
        boxf 20,50,280,70
        boxf 20,75,280,95
        boxf 20,100,280,120
        boxf 20,125,280,145
        repeat 5                              ;今回結果は赤表示
          j=cnt+1
          if cnt=jyun:color 255,0,0:else:color 0,0,0
          pos 40,cnt*25+30:mes " "+j+". "+rank.cnt+" 単語クリア"
        loop
;
        mes " "
        font "MS ゴシック",14
        color 255,255,0
        mes "       <入力スピード>"
        mes " "
        mes " "+gtm+" 秒で "+gko+" 字入力できました"
        mes " よってスピードは "+fun+" 字/分です"
        mes " "
        color 255,0,0
        mes "     <ここまでの合計結果>"
        mes " "
        mes " "+kaigou+" ラウンド挑戦しました"
        mes " "+secgou+" 秒で "+mojgou+" 字入力できました"
        mes " 平均スピードは "+fungou+" 字/分です"
        color 128,255,255
```

★大雪の夜に停電。部屋，誰かいる！ライトフリーズ！／アイスコーヒーを飲んだ後にサイダーを飲むと，メッコーラの味がする。／やるなDr.! じつは私も返信。北の方角に影さんを置くと。（愛知県名古屋市・**最馬場可？歳**）……**【影：書けばいいってものでも…。】**

```
	font "ＭＳ ゴシック",18
	pos 15,330
	mes "続ける：[Space]　やめる：[Esc]"
*tykey
	wait 1:stick key,0:if key=0:goto *tykey
	if key=128:goto *tend		;やめる
	gsel 2,-1:goto *thajime		;続ける
;
;======== サブルーチン ===========================
;タイプミス時のイベント
*ttigau
	snd 4	;ちがうよ効果音　うるさかったらはずして
	if sayuu=1:pos 16,320:else:pos 296,320
	gcopy 5,0,0,128,64	;訂正してねの画像
	return
;
;サルが落ちる処理
*tsaru
	redraw 0
	color 128,128,0:boxf 190,10,280,290 ;サルの通路
	color 128,64,0:boxf 180,290,290,310 ;地面
	color 128,128,0:boxf 450,10,540,290 ;サルの通路右
	color 128,64,0:boxf 440,290,550,310 ;地面右
	syy1=syy1+2:syy2=syy2+2		;落下距離
	pos 200,syy1:gcopy 6,0,0,60,68 ;サルを表示
	pos 470,syy2:gcopy 6,0,0,60,68 ;サルを表示右
	redraw 1
	if (syy1>=225)|(syy2>=225):dame=1 ;落ちたゲームオーバーフラグオン
	return
;
;元に戻るサル
*tagaru1
	syy1=10
	redraw 0
;
	color 128,128,0:boxf 190,10,280,290 ;サルの通路
	color 128,64,0:boxf 180,290,290,310 ;地面
	pos 200,syy1:gcopy 6,0,0,60,68	;サルを表示
	redraw 1
	return
;
*tagaru2
	syy2=10
	redraw 0
;
	color 128,128,0:boxf 450,10,540,290 ;サルの通路右
	color 128,64,0:boxf 440,290,550,310 ;地面右
	pos 470,syy2:gcopy 6,0,0,60,68	;サルを表示右
	redraw 1
	return
;
;クリア数評価イベント
*thyouka
	color 0,0,0		;メッセージ用ボックス開く，ように見せる
	repeat 20		;要はもったいつける
	  y=cnt*5+10
	  boxf 20,90,140,90+y
	  wait 2
	loop
	color 255,255,255:font "ＭＳ ゴシック",24,1
	pos 25,110
	if (gtan>=0)&(gtan<=10):mes "もしもし":mes "ねてる？"
	if (gtan>10)&(gtan<=20):mes "もう少し":mes "がんばろ"
	if (gtan>20)&(gtan<=30):mes "まあまあ":mes "じゃない"
	if (gtan>30)&(gtan<=40):mes "なかなか":mes "やるじゃ":mes "な～い"
	if (gtan>40)&(gtan<=50):mes "すごいよ":mes "タイプの":mes "達人だね"
	if gtan>50		:mes "卒業だよ":mes "免許皆伝":mes "神業の域"
	return
;
;時間測定計算
*tgettime1
	gettime tm1,4
	gettime tm2,5
	gettime tm3,6
	return
;
*tgettime2
	gettime etm1,4
	gettime etm2,5
	gettime etm3,6
	ts=tm1*3600:ts=tm2*60+ts:ts=tm3+ts
	te=etm1*3600:te=etm2*60+te:te=etm3+te
	tk=te-ts				;１単語入力時間(秒)
	gtm=gtm+tk:secgou=secgou+tk		;合計　秒数加算
	return
;
;データファイルload 配列へsave
*tread
	sdim aaa,7200
	bload "english.txt",aaa	;英語テキスト読み込み
	notesel aaa
	sdim bbb,7200
	notemax mmm
	repeat mmm
	  noteget bbb,cnt
	  eig.cnt=bbb		;英語配列バッファへ
	loop
;
	sdim aaa,7200
	bload "roman.txt",aaa	;ローマ字テキスト読み込み
	notesel aaa
	sdim bbb,7200
	notemax mmm
	repeat mmm
	  noteget bbb,cnt
	  rom.cnt=bbb		;ローマ字配列バッファへ
	loop
;
	sdim aaa,7200
	bload "japanese.txt",aaa ;日本語テキスト読み込み
	notesel aaa
	sdim bbb,7200
	notemax mmm
	repeat mmm
	  noteget bbb,cnt
	  jap.cnt=bbb		;日本語配列バッファへ
	loop
	return
;
;プログラムEND
*tend
	end
;;================================//EOD
```

★ついに受験シーズン突入です。逆風に負けず投稿に励もうと思います（っておい！）／コンピュータウイルスと戦ってる夢を見ました。「こいつはもう手遅れだ！」「とうっ！」「ズガァァン」……結局パソ破。（東京都練馬区／**ルストフレム15歳**）……【**影：ごくろう！**】

《リスト 2》english.txtのリスト

```
friend
mother
father
girl
boy
japanese
kind
teacher
dog
cat
hair
help
today
yesterday
car
young
water
great
fish
horse
train
sunday
nurse
doctor
now
animal
park
airplane
music
beautiful
library
bird
snake
monkey
pig
school
give
earth
baby
house
run
walk
think
know
face
eye
ear
work
understand
river
mountain
snow
rain
red
white
black
blue
right
left
bus
sleep
flower
rose
money
baseball
winter
summer
movie
tea
lunch
box
tiger
rabbit
chair
desk
book
pencil
apple
orange
milk
banana
hotel
dictionary
mirror
head
history
science
future
forget
sugar
egg
room
answer
melon
sing
peace
wife
husband
english
knife
garden
son
daughter
believe
village
dream
tennis
window
drink
sea
remember
swim
west
east
south
north
easy
world
moon
spring
important
buy
read
student
live
sky
famous
blood
write
useful
body
television
stone
friday
ticket
umbrella
cloud
map
lion
policeman
magazine
uncle
aunt
examination
diary
salt
leaf
bicycle
potato
april
toy
elephant
fact
fruit
tooth
cow
hobby
mathematics
difficult
weather
bank
expensive
ship
hungry
short
long
strong
sit
child
enjoy
conversation
secret
respect
poem
classmate
piano
novel
busy
morning
problem
season
tomorrow
wait
question
night
culture
power
interesting
country
together
brother
sister
love
tuesday
monday
january
november
green
hand
door
fast
station
example
tired
hospital
popular
wash
experience
kitchen
machine
climb
fire
true
mistake
september
touch
wind
favorite
radio
mouth
careful
nose
early
grow
law
promise
foreigner
million
passenger
government
information
neighber
museum
hundred
thousand
medicine
scissors
refrigerator
ambulance
```

《リスト 3》japanese.txtのリスト

```
友達
母
父
女の子
男の子
日本語
親切な
先生
犬
猫
髪の毛
助ける
今日
昨日
車
若い
水
偉大な
さかな
馬
汽車
日曜日
看護婦
医者
今
動物
公園
飛行機
音楽
美しい
図書館
鳥
蛇
猿
豚
学校
与える
地球
赤ちゃん
家
走る
歩く
思う
知る
顔
目
耳
働く
理解する
川
山
雪
雨
赤い
白い
黒い
青い
右
左
バス
眠る
花
ばら
お金
野球
冬
夏
映画
紅茶
昼食
箱
虎
うさぎ
いす
机
本
鉛筆
りんご
オレンジ
牛乳
バナナ
ホテル
辞書
鏡
頭
歴史
科学
未来
忘れる
砂糖
たまご
部屋
答える
メロン
歌う
平和
妻
夫
英語
ナイフ
庭
息子
娘
信じる
村
夢
テニス
窓
飲む
海
思い出す
泳ぐ
西
東
```

★忙しいときに限って遊びたくなりませんか？／最近はグルグルに目覚めました／カレンダーは1月7日ぐらいになると半額になっている／牛丼のつぎはカルビ丼ですか？／腰のひねりを忘れるな，アミーゴ！（東京都練馬区・**どらねく15歳**）……**【影：まいった！】**

```
南
北
易しい
世界
月
春
大切な
買う
読む
学生
住む
空
有名な
血
書く
役に立つ
体
テレビ
石
金曜日
切符
傘
雲
地図
ライオン
警官
雑誌
おじ
おば
試験
日記
塩
葉
自転車
じゃがいも
四月
おもちゃ
象
事実
くだもの
歯
牝牛
```

```
趣味
数学
むずかしい
天気
銀行
高価な
船
空腹な
短い
長い
強い
座る
子供
楽しむ
会話
秘密
尊敬する
詩
同級生
ピアノ
小説
忙しい
朝
問題
季節
明日
待つ
質問
夜
文化
力
おもしろい
国
いっしょに
兄弟
姉妹
愛する
火曜日
月曜日
一月
十一月
緑の
```

```
手
ドア
速い
駅
例
疲れた
病院
人気のある
洗う
経験
台所
機械
登る
火
本当の
間違い
九月
触れる
風
お気に入りの
ラジオ
口
注意深い
鼻
早く
成長する
法律
約束
外国人
百万
乗客
政府
情報
隣人
博物館
百
千
薬
はさみ
冷蔵庫
救急車
```

《リスト 4》roman.txtのリスト

```
tomodati
haha
titi
onnnanoko
otokonoko
nihonngo
sinnsetuna
sennsei
inu
neko
kaminoke
tasukeru
kyou
kinou
kuruma
wakai
```

```
mizu
idaina
sakana
uma
kisya
nitiyoubi
kanngofu
isya
ima
doubutu
kouenn
hikouki
onngaku
utukusii
tosyokann
```

```
tori
hebi
saru
buta
gakkou
ataeru
tikyuu
akatyann
ie
hasiru
aruku
omou
siru
kao
me
mimi
```

```
hataraku
rikaisuru
kawa
yama
yuki
ame
akai
siroi
kuroi
aoi
migi
hidari
basu
nemuru
hana
bara
okane
yakyuu
fuyu
natu
eiga
koutya
tyuusyoku
hako
tora
usagi
isu
tukue
honn
ennpitu
rinngo
orennji
gyuunyuu
banana
hoteru
jisyo
kagami
atama
rekisi
kagaku
mirai
wasureru
satou
tamago
heya
kotaeru
meronn
utau
heiwa
tuma
otto
eigo
naifu
niwa
musuko
musume
sinnjiru
mura
yume
tenisu
mado
nomu
umi
omoidasu
```

```
oyogu
nisi
higasi
minami
kita
yasasii
sekai
tuki
haru
taisetuna
kau
yomu
gakusei
sumu
sora
yuumeina
ti
kaku
yakunitatu
karada
terebi
isi
kinnyoubi
kippu
kasa
kumo
tizu
raionn
keikann
zassi
oji
oba
sikenn
nikki
sio
ha
jitensya
jyagaimo
sigatu
omotya
zou
jijitu
kudamono
ha
meusi
syumi
suugaku
muzukasii
tennki
ginnkou
koukana
fune
kuufukuna
mijikai
nagai
tuyoi
suwaru
kodomo
tanosimu
kaiwa
himitu
sonnkeisuru
si
doukyuusei
```

```
piano
syousetu
isogasii
asa
mondai
kisetu
asita
matu
situmonn
yoru
bunnka
tikara
omosiroi
kuni
issyoni
kyoudai
simai
aisuru
kayoubi
getuyoubi
itigatu
jyuuitigatu
midorino
te
doa
hayai
eki
rei
tukareta
byouinn
ninnkinoaru
arau
keikenn
daidokoro
kikai
noboru
hi
honntouno
matigai
kugatu
fureru
haze
okiniirino
rajio
kuti
tyuuibukai
hana
hayaku
seityousuru
houritu
yakusoku
gaikokujinn
hyakumann
jyoukyaku
seifu
jyouhou
rinnjinn
hakubutukann
hyaku
senn
kusuri
hasami
reizouko
kyuukyuusya
```


★OFは全ページにならないのですか。(岐阜県加茂郡・ふっとちょ27歳) ……【**影**：それだけは勘弁して！お願い！**編**：仕事中に寝てると思ったら，うなされはじめましたよ。**影**：うわーっ！ハガキの山に押しつぶされる〜。**つぐ美**：寝ててもうるさい影さんです。】

Windows(HSP V2.5)用



回転するパネルに目を凝らせ

Rolling Panels

（ローリング パネルズ）

斎藤　和哉（28歳）

内容

回転するパネルの中から，中心に表示されているパネルと同じものを探すゲームです。

操作はマウスのみを使用します。中心に表示されているパネルと同じパネルにマウスカーソルを合わせて，左クリックするだけの単純なゲームです（写真2）。

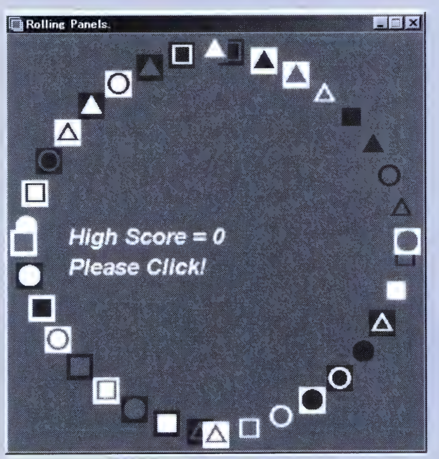
遊び方

起動すると，タイトル(?)が出ますので，左クリックでゲームスタートです(右クリックすると，ゲームを終了します)。

ゲームがスタートすると，ステージ番号が出てからパネルが広がります。この時，マウスカーソルは中心部に移動します。

中心に表示されているパネルと同じものを見つけて，左クリックします（写真）。同じものをクリックできればステージクリア。ステージスコアが加算されます。ステージスコアは，時間が経つごとに制限無く減っていきますので，できるだけ早く探す必要があります。

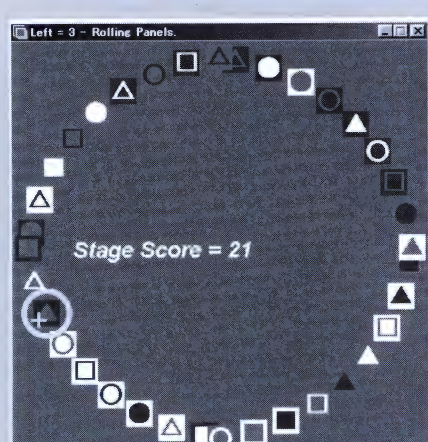
クリックした所に，中心と同じパネル



《写真1》回転するパネルを利用したゲーム



《写真2》真ん中に表示されるパネルと同じものを探せ



《写真3》パネルを探し当てたら即クリックしろ



《写真4》3回ミスをするとゲームオーバー

★さらなる刷新の予感。期待をこめて。WinXPはまだ危険なのでサービスパックを待ちましょう！（福井県武生氏・つかちー17歳）
……【編：2002年1月号，2月号となかなか評判がいいんですよね。学校の図書館に置いてもらうよう，読者も働きかけてください。】

が無かった場合は，ミスになります。3回ミスをすると，ゲームオーバー（写真4）。

なお，ステージが進むと，回転速度が速くなります。

●プログラムの仕組み●

非常に単純な構造になっています。

①パネルを回転させるための，テーブルを作ります。
②プレートの画像を作成します。
③タイトルを表示します。
④ゲームが始まると，パネルを回しながら，クリックを監視します。
⑤クリックした所が正解のパネルかどうかを判定し処理を振りわけます。

パネルの回転は，サブルーチンで処理しています

HSP標準のエディタや，適当なテキストエディタで入力し，適当な名前(*.as)

●プログラムの入力方法，注意点●

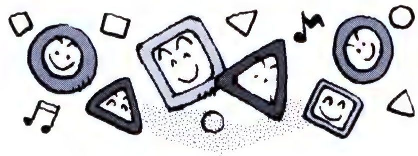
で保存してください。外部リソースを一切使用していないので，スクリプトソースだけで動作します。

ソースの行数を減らすため，詰めたソースになっています。入力しづらいかもしれません。

Windowsの標準パレットのみを使用しているため，色をいじる場合には，パ

● 改造のアドバイス ●

レットの管理に注意する必要があります。



CHECKER FLAG

影：一見単純だけど，けっこう面白い。後半のステージになってくるとパネルの回転がかなり速くなってくるので，プレイしていても単調に感じないゲームだな。

Dr.D：ゲームの内容以上に注目してほしいのがリストじゃ。このリストは非常にコンパクトに組まれており，最近の投稿作品の中でも群を抜いているリストと言える。このゲームの肝となる円に沿ったパネルの表示は，説明にもあるようにパネルの表示位置を決める処理を“*_output”でルーチンとしてまとめているため，凝った表現ながらもリストがシンプルにまとまる。また，この表現を行うのに必須となる三角関数(sin，cos)の値を，リストの最初の方で自分で作成している点にも注目じゃな。

編：各ステージ開始時にマウスカーソルが中央に移動するのが，チョットしたこだわりだなっと感じました。

入力方法は33ページをご覧ください。

《リスト》Rolling Panelsのプログラム・リスト

```
;------------------------------------------------------------
;  Rolling Panels. / By SAITOH Kazuya --- http://sslabo.lib.net/asg/
;------------------------------------------------------------
dim st, 72 : dim ct, 72 : dim x, 36 : dim pl, 36
randomize : ttl = "Rolling Panels."
onexit *_quit

; --- [ 第1象限分の sin テーブル ]
st=0,89,178,265,350,433,512,587,658,724,784,839,887,928,962,989,1008,1020

; --- [ sin, cos テーブルを作成 ]
repeat 18 : i = 35 - cnt : st.i = st.cnt : loop;  第2象限
repeat 36 : i = 36 + cnt : st.i = -st.cnt : loop;  第3,4象限
repeat 72 : i = 18 + cnt : if i>71 : i -= 72
        ct.cnt = st.i : loop;          cos テーブルの作成

; --- [ プレートの作成 ]
max = 0 : fname = "ＭＳ ゴシック" : fname2 = "Arial";       初期設定
buffer 2,440,440,1 : color 128,128,128 : boxf 0,0,439,439
buffer 3,1008,28,1 : pch = "○△□●▲■"
color 0,0,255 : boxf 0,0,335,27 : color 255,0,0 : boxf 336,0,671,27
color 255,255,255 : boxf 672,0,1007,27 : font fname,27,1
pos 0,0 : mes pch : pos 336,0 : mes pch
color 0,0,255 : pos 504,0 : mes pch + pch
color 255,0,0 : pos 168,0 : mes pch : pos 840,0 : mes pch

repeat 36 : x.cnt = cnt * 28 : loop

; --- [ Title ]
*_title
screen 0,440,440,1,(dispx - 446) / 2,(dispy - 460) / 2 : title ttl
gmode 1,28,28 : font fname2,24,3 : objsize ,,32 : color 0,0,0
gosub *_shuffle
m = 204 : b = 0 : ans = -1
repeat
        repeat 5 : wait 4 : stick i,,1 : if i&768 : break
        loop : if i==512 : goto *_quit
        if i==256 : break
        b++ : if b > 71 : b = 0
        gosub *_output
        color 255,255,255 : pos 64,200
        mes "High Score = "+max : mes "Please Click!"
loop

; --- [ 始 ]
stage = 1 : left = 3 : score = 0
```

★ベーマガマニアとして電波新聞社のオリジナル・カレンダーはぜひともいただきたい一品ですな。（愛媛県東宇和郡・毛利祐允18歳）
……【**編**：2月号で行ったプレゼントのことですね。**つぐ美**：カレンダー希望者結構多いですね。激戦かもしれませんよ。】

```
*_stage_start
rnd ans, 36 : title "Left = "+left+" - "+ttl
wt = 80 - ((stage / 2) * 10) : if wt<20 : wt = 20
bp = stage * 10 + 15
pos 0,0 : gcopy 2,0,0,440,440
color 255,255,255 : pos 160,208 : mes "Stage  "+stage : wait 100

; --- [ 広がる ]
gosub *_shuffle : b = 51
repeat 21 : wait 4 : m = cnt * 10 + 4
        gosub *_output : b++ : loop

ginfo 3 : mouse prmx - 223, prmy - 223
b = 0 : gosub *_output

; --- [ Game Loop ]
*_game_loop
repeat 5 : await wt : stick i,,1 : if i&256 : break
loop : if i==256 : goto *_click
b++ : if b > 71 : b = 0
gosub *_output
bp-- : goto *_game_loop

*_click;
mx = mousex : my = mousey
repeat 36 : if pl.cnt==ans : i = cnt : break
loop
a = i * 2 + b : if a > 71 : a -= 72
a = 71 - a : px = (m*st.a)/1024+206 : py = 206-((m*ct.a)/1024)

color 0,255,0 : font fname,64 : pos px - 18, py - 18
boxf mx-1,my-8,mx+1,my+8 : boxf mx-8,my-1,mx+8,my+1

i = 2 : a = 28 + i; 許容誤差
if (mx<(px-i))|(my<(py-i)) : goto *_miss
if ((mx-px)>a)|((my-py)>a) : goto *_miss

mes "○" : pos 206,206 : gcopy 2,0,0
font fname2,24,3 : color 255,255,255 : pos 64,208
mes "Stage Score = "+bp : wait 100
score += bp;         点数を足す
mes "Total Score = "+score : wait 200
stage++ : goto *_stage_start

*_miss
mes "□" : pos 206,206 : gcopy 2,0,0
font fname2,24,3 :color 255,255,255 : pos 64,208
mes "Miss !!": wait 100 : left--
if left>0 : goto *_stage_start
mes "Game Over !!" : mes "Total Score = "+score : wait 200
if score>max : max = score

color 128,128,128
repeat 55
        i = cnt * 8 : boxf 0,i,439,i+7 : boxf 0,444-i,439,439-i
        await 20
loop
goto *_title

;-----------------------------------------------
; --- [ 混ぜる ]
*_shuffle
repeat 36 : pl.cnt = cnt : loop
repeat 36 : rnd a, 36 : b = pl.cnt : pl.cnt = pl.a : pl.a = b : loop
return

; --- [ 表示 ]
*_output
gsel 0 : redraw 2 : pos 0,0 : gcopy 2,0,0,440,440
repeat 36
        a = cnt * 2 + b : if a > 71 : a -= 72
        a = 71 - a
        pos (m*st.a)/1024+206,206-((m*ct.a)/1024) : a = pl.cnt
        gcopy 3,x.a,0
loop
if ans>=0 : pos 206,206 : gcopy 3,x.ans,0
redraw 1 : return

*_quit
mouse dispx / 2, (dispy / 2) + 32 : dialog "End.",0,ttl : end
;-----------------------------------------------
;----- [ EOF ]
```

★ベーマガのホームページに投稿プログラムのキーワードによる検索コーナーを設けてください（プログラム名はわかっていたのですが，何月号か忘れてしまったので）。お願いします。（埼玉県北本市・**有馬琢也27歳**）……【**編集長**：いい意見だ。影くん，まかせた。】

VB投稿の名作をiアプリへ移植



iエジプトの石（いし）

サル吉（16歳）

内容

　このゲームは，本誌2001年10月号に掲載された『エジプトの石』の移植版です。
　遊び方も『エジプトの石』とほとんど同じで，時間内に同じ石をペアで消していき，すべて消すというゲームです。

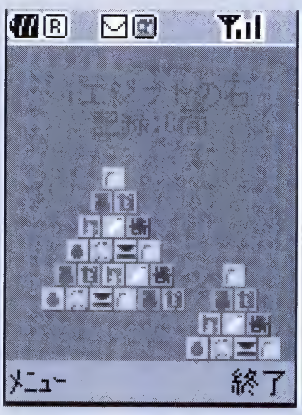
ルール

①同じ石をぶつけると消えます。
②すべての石を消すとクリアです。
③石は動き出すと，他の石に当たるまで動き続けます。同じ模様の石に当たると二つとも消えます。
④石がテーブルから飛び出すとミスです。
⑤その面の時間制限内にクリアしなければなりません。クリアすると，石が二つ，時間が４秒ずつ増えていきます。

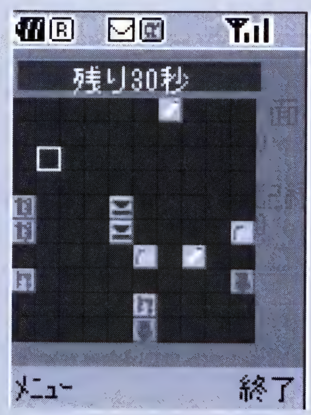
操作方法

　主に，カーソルキーと決定ボタンで操作します。また，カーソルキーは上下左右それぞれ２，８，４，６に，決定キ

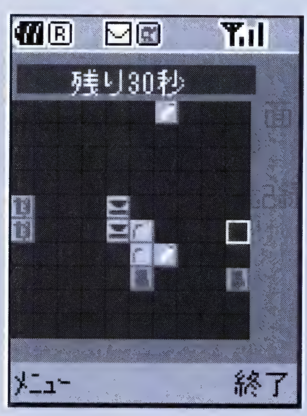
入力方法は33ページをご覧ください。



《写真１》iアプリで作られたエジプトの石



《写真２》画面構成はＶＢ版の雰囲気そのまま



《写真３》iアプリ用にアレンジされた操作も快適



《写真４》メニュー表示にこだわりがある



左から0.gif～9.gifの画像

CHECKER FLAG

つぐ美：今までもポケコンなどの投稿作品をiアプリへ移植した作品はありましたが，今回はVBの投稿作品の移植版です。画面範囲が限られるiアプリでありながら，画面構成がVB版そのままなのが驚きですね。完全移植に近い出来と言えるんじゃないかしら？
影：ただVB版と同様，制限時間の設定がキビシイなぁ。まぁ，これは各自で改造してもらうという手もあるな。ソフト側から設定できるようにしてもいいかも？
Dr.D：VB版と違うのは，iアプリに合わせた操作方法とメニュー画面の表示などじゃな。ただ，このメニューはキレイに表示されるのじゃが，チト効率が悪い形で実現しているのが残念。基本的なやり方で決して悪い方法ではないのじゃが，この部分を配列をうまく用いるなど工夫すれば，リストの簡略化につながりそうな感じじゃな。

★言語の種類が多いのですが，言語によって得意・不得意があるのでしょうか？（静岡県清水市・瑠璃27歳）……【**編**：いま，各言語の特徴をまとめている段階です。近いうちに記事にしますのでお楽しみに！**つぐ美**：初めての人は，言語選択はは悩みどころですよね。】

《リスト２》iEgyptScrren.javaのリスト

```
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import javax.microedition.io.*;
import java.util.*;
import java.io.*;

//■■■■■■ メインプログラム ■■■■■■
final class iEgyptScreen extends Canvas{
        public void paint(Graphics g){}
        int Game=0,First;           //ゲームの状態,初回プレイか確認
        int cx,cy,cc,mx,my,mh,Move;//カーソルの状態(x,y,c),動くブロックの座標方向(x,y,h)
        int Level,HiLevel;          //レベル,最高レベル
        int timer,timer2,Timetemp; //タイマー
        int R,G,B,backL;            //背景色(r,g,b),バックライト
        int Total,menu,load;        //スコアのトータル,メニュー
        int Width,Height;           //画面の大きさ
        int i,j,l,temp;             //雑用変数
        int Map[][]     =new int[12][12];//メイン画面
        int Maptemp[][]=new int[12][12];//リトライ用
        int Time[]      =new int[40];    //各面のクリアタイム
        int dat[]       =new int[10];    //保存用
        Image    img[] =new Image[10];   //イメージ格納
        Random   rnd    =new Random();   //乱数列を作成
        Dialog   Dlg1   =new Dialog(Dialog.DIALOG_YESNO,"確認");//確認YESNOダイアログ
        Graphics g      =getGraphics();

        //■■■■■■ニューゲーム関数■■■■■■
        public void NewGame(){
                int x,y,m,n,g,f,block;

                load=0;
                Game=10;

                cx=cy=1;cc=Move=0;
                Timetemp=30+(Level-6)*4;//タイマーの設定
                timer=timer2=0;
                //マップの初期化
                for(i=0;i<12;i++)
                        for(j=0;j<12;j++)
                                if(i==0||i==11||j==0||j==11)Map[i][j]=20;
                                else Map[i][j]=0;
                //ブロックの配置
                Map[5][5]=Map[6][5]=1;
                for(i=2;i<=Level;i++){
                        g=f=0;
                        block=i%9;
                        if(block==0)block=9;
                        while(true){
                                m=Math.abs(rnd.nextInt()%8)+2;
                                n=Math.abs(rnd.nextInt()%8)+2;
                                if(Level<12){
                                        m=Math.abs(rnd.nextInt()%6)+3;
                                        n=Math.abs(rnd.nextInt()%6)+3;
                                }
                                g++;
                                if(g>100){f=1;break;}
                                do{
                                        x=Math.abs(rnd.nextInt()%2);
                                        y=Math.abs(rnd.nextInt()%2);
                                }while(Math.abs(x+y)!=1);
                                if(Map[m][n]==0&&Map[m+x][n+y]==0){
                                        if(Map[m-1][n]>0||Map[m+1][n]>0||Map[m][n-
1]>0||Map[m][n+1]>0){
                        Map[m][n]=Map[m+x][n+y]=block;break;
```

ーは５に対応しています。

まず，動かしたい石の上にカーソルを持っていきます。次に決定ボタンを押すとカーソルの色が赤色に変わります。その状態で動かしたい方向のキーを押すと，その方向へ移動していきます。移動が終わると，カーソルの色が黄色に戻ります。カーソルの色が赤色の状態(選択状態)の時は，カーソルの移動はできません。また，選択状態を解除したい場合は，もう一度決定キーを押すと黄色のカーソルに戻ります。

プログラムの仕組み

このプログラムは，本誌2001年12月に掲載された『iSCROPE』を参考にさせていただきました。

ゲームは主に，11個の画面からなっています。Gameという変数をswitchで振りわけて描画，キー操作を行っています。

タイマーは，常に１秒ごとにカウントされています。また，TimeCheck()という関数は，タイマーが前のループの時から動いていれば１を，動いていなければ０を返す仕組みになっています。

プログラムの入力方法

10×10の大きさの画像データを10個用意します。あと，２つのソースファイルを入力してコンパイルするとできます。

注意点

ゲームは「終了」というボタンを押さないとデータは保存されません。「はじめから」という操作を行っても，「終了」というボタンを押すまでは，セーブデータはクリアされません。

一部の機種では，面作成に大変時間がかかりますが，ご了承ください。

改造のアドバイス

やっぱり面作成でしょう。今の方法は，『エジプトの石』のソースをそのまま使っているので，ここを改造して快適に楽しめるようにすればいいと思います。

《リスト１》iEgypt.javaのリスト

```
d import com.nttdocomo.ui.*;

public final class iEgypt extends IApplication
{
        public void start(){
                iEgyptScreen screen=new iEgyptScreen();
                Display.setCurrent(screen);
                ShortTimer t=ShortTimer.getShortTimer(screen,0,1000,true);
                t.start();
                screen.exe();
        }
}
```

★大学入試を控えBASIC対策が組まれていると耳にし，久しぶりに手にとりました。見ないうちにｉアプリまで扱っているのに驚いた反面，予備校のスタッフまでもが懐かしむほど変わらないスタイルにほっとしました。（神奈川県横浜市・Gleam18歳）……【影：は。】

```
                        }
                    }
                }
                if(f==1)break;
                NewGameSub(m,n);
            }

            for(i=1;i<=30;i++){
                do{
                    m=Math.abs(rnd.nextInt()%10)+1;
                    n=Math.abs(rnd.nextInt()%10)+1;
                }while(Map[m][n]==0);
                NewGameSub(m,n);
            }
            //リトライ用に保存
            for(i=0;i<12;i++)
                for(j=0;j<12;j++)
                    Maptemp[i][j]=Map[i][j];
            load++;
        }

        //■■■■■■ニューゲーム関数(2)■■■■■■
        public void NewGameSub(int m,int n){
            int x,y,m2=0,n2=0,block=0;

            for(l=1;l<=20;l++){
                do{
                    x=1-Math.abs(rnd.nextInt()%3);
                    y=1-Math.abs(rnd.nextInt()%3);
                }while(Math.abs(x+y)!=1);
                if(Map[m-x][n-y]>0&&Map[m-x][n-y]<10&&Map[m][n]!=Map[m-x][n-
y]&&Map[m+x][n+y]==0){
                    block=Map[m][n];m2=m;n2=n;
                    do{
                        m+=x;n+=y;
                    }while(Map[m][n]==0);
                    if(Map[m][n]!=block){
                        Map[m2][n2]=0;Map[m-x][n-y]=block;break;
                    }
                }
            }
            load++;
        }

        //■■■■■■リトライ関数■■■■■■
        public void ReTry(){
            cx=1;cy=1;cc=0;Move=0;
            Timetemp=30+(Level-6)*4;
            timer=0;timer2=1;
            //マップのコピー
            for(i=0;i<12;i++)
                for(j=0;j<12;j++)
                    Map[i][j]=Maptemp[i][j];
            Game=1;
        }

        //■■■■■■ピラミッドの作成■■■■■■
        public void Pira(int height,int x,int y){
            temp=0;
            for(i=0;i<height;i++)
                for(j=-i;j<=i;j+=2){
                    temp++;
                    g.drawImage(img[temp%9+1],j*5+x,i*10+y);
                }
        }

        //■■■■■■セーブデータのロード■■■■■■
        public void Load(){
            try{
                    DataInputStream
in=Connector.openDataInputStream("scratchpad:///0");
                    for(i=0;i<6;i++)dat[i]=in.readInt();
                    First  =dat[0];
                    HiLevel=dat[1];
                    if(HiLevel==0)HiLevel=6;
                    backL  =dat[2];
                    R      =dat[3];
                    G      =dat[4];
                    B      =dat[5];
                    for(i=0;i<40;i++)Time[i]=in.readInt();
                    in.close();
            }catch(IOException e){}
            //バックライトの設定
            if(backL==1)PhoneSystem.setAttribute(0,1);//ON
            else PhoneSystem.setAttribute(0,0);//OFF
            if(First==0){R=G=160;B=0;First=1;}
        }

        //■■■■■■セーブデータのセーブ■■■■■■
        public void Save(){
            try{
                DataOutputStream
out=Connector.openDataOutputStream("scratchpad:///0");
                dat[0]=First;dat[1]=HiLevel;dat[2]=backL;
                dat[3]=R;dat[4]=G;dat[5]=B;
                for(i=0;i<6;i++)out.writeInt(dat[i]);
                for(i=0;i<40;i++)out.writeInt(Time[i]);
                out.close();
            }catch(IOException e){}
        }

        //■■■■■■タイマーのチェック(1秒たったら1を返す)■■■■■■
        int TimeCheck(){
            if(timer2!=timer){timer2=timer;return 1;}
            return 0;
        }

        //■■■■■■メイン関数■■■■■■
        void exe(){
            int iKey;//キー
            Font font;

            //画面の大きさの取得
            Width =getWidth();
            Height=getHeight();

            //文字の大きさの設定
            String platform=System.getProperty("microedition.platform");
            if(platform.equals("SO503i")||platform.equals("D503i"))
                font=Font.getFont(Font.SIZE_SMALL);
            else font=Font.getDefaultFont();
            g.setFont(font);

            Load();//セーブデータのロード

            //ソフトキーの文字設定
            setSoftLabel(SOFT_KEY_1,"ﾒﾆｭｰ");
            setSoftLabel(SOFT_KEY_2,"終了");

            //□□□□□□画像ファイル読み込み□□□□□□
            MediaImage mi;
            try{
```

★毎月，読者からGET＆RUNへの質問の解決法を募集するってのはどうでしょう。（石川県金沢市・**村田春佳17歳**）……【**Dr.D**：なるほど。読者は質問者であり，回答者ってことか。これはおもしろい！試しにやってみるか？**影**：そのページ，私が担当ってこと？】

```
for(i=0;i<10;i++){
        mi=MediaManager.getImage("resource:///"+i+".gif");
        mi.use();
        img[i]=mi.getImage();
    }
}catch(Exception e){}

//□□□□□□□ 無限ループ□□□□□□□
while(true){
    iKey=getKeypadState();//キー入力
    if(iKey==(1<<Display.KEY_SOFT2)){
        Save();//データセーブ
        IApplication.getCurrentApp().terminate();//終了
    }
    //◆◆◆◆◆◆ ブロックが動いているなら動かす◆◆◆◆◆◆
    if(Move==1){
        switch(mh){
        case 0:i=my-1;j=mx;break;//上
        case 1:i=my;j=mx+1;break;//右
        case 2:i=my+1;j=mx;break;//下
        case 3:i=my;j=mx-1;break;//左
        }
        if(Map[i][j]==0){
            Map[i][j]=Map[my][mx];Map[my][mx]=0;
            switch(mh){
            case 0:my--;break;
            case 1:mx++;break;
            case 2:my++;break;
            case 3:mx--;break;
            }
        }
        else if(Map[i][j]==20){
            menu=0;Move=0;cc=0;Game=3;
        }
        else if(Map[i][j]!=Map[my][mx]){
            Move=0;cc=0;
        }
        else if(Map[i][j]==Map[my][mx]){
            Map[i][j]=Map[my][mx]=0;Move=0;cc=0;
        }
    }

    //◆◆◆◆◆◆ クリア確認◆◆◆◆◆◆
    if(Game==1){
        for(i=1,l=0;i<11;i++)
            for(j=1;j<11;j++)
                if(Map[i][j]!=0) l++;
        if(l==0){menu=0;Game=2;}
    }
    //◆◆◆◆◆◆ ゲームオーバー確認◆◆◆◆◆◆
    if(Game==1&&Timetemp==0){menu=0;Game=3;}

  //◆◆◆◆◆◆ソフトキー1の文字設定◆◆◆◆◆◆
  if(Game==5)setSoftLabel(SOFT_KEY_1,Integer.toString(temp+Level-6)+"面");
  else setSoftLabel(SOFT_KEY_1,"ﾒﾆｭｰ");

    //◆◆◆◆◆◆ 画面出力◆◆◆◆◆◆
    g.lock();//画面のロック
    //画面を(R,G,B)で初期化
    g.setColor(g.getColorOfRGB(R,G,B));
    if(Game!=4&&Game!=5)g.fillRect(0,0,Width,Height);
    switch(Game){
    case 0://タイトル
        g.setColor(g.getColorOfRGB(255-R,255-G,255-B));
        g.drawString("iエジプトの石",24,24);
        g.drawString("記録:"+Integer.toString(HiLevel-6)+"面",36,36);
        g.setColor(g.getColorOfRGB(255,0,0));
        Pira(6,40,50);Pira(4,90,90);
        g.setColor(g.getColorOfRGB(0,255,0));
        break;
    case 1://ゲーム中
        //主画面描画
        for(i=1;i<11;i++)
            for(j=1;j<11;j++)
                g.drawImage(img[Map[i][j]],j*10-
10,i*10+10);

        //□を描画
        if(cc==0)g.setColor(g.getColorOfRGB(255,255,0));
        else g.setColor(g.getColorOfRGB(255,0,0));
        g.drawLine(cx*10-10,   cy*10+10,   cx*10+9-10,cy*10+10);
        g.drawLine(cx*10+9-10,cy*10+10,   cx*10+9-10,cy*10+9+10);
        g.drawLine(cx*10+9-10,cy*10+9+10,cx*10-10,   cy*10+9+10);
        g.drawLine(cx*10-10,   cy*10+9+10,cx*10-10,   cy*10+10);
        //タイマー設定&表示
        if(TimeCheck()==1)Timetemp--;
        if(Timetemp>5)g.setColor(g.getColorOfRGB(0,0,255));
        else g.setColor(g.getColorOfRGB(255,0,0));
        g.fillRect(2,5,Timetemp*100/(30+(Level-6)*4),12);
        g.setColor(g.getColorOfRGB(255,255,0));
        g.drawString("残り"+Timetemp+"秒",25,17);
        g.setColor(g.getColorOfRGB(255-R,255-G,255-B));
        g.drawString("  面",100,30);
        g.drawString(Integer.toString(Level-6),100,42);
        g.drawString("記録",100,64);
        g.drawString(Integer.toString((100-Time[Level-
6]*100/(30+(Level-6)*4))%100),100,76);
        break;
    case 2://ステージクリア
        g.setColor(g.getColorOfRGB(255-R,255-G,255-B));
        g.drawString("くりあー♪",35,40);
        g.fillRect(12+menu*36,84,24,12);
        g.setColor(g.getColorOfRGB(0,0,255));
        if(Time[Level-6]>30+(Level-6)*4-Timetemp||Time[Level-
6]==0){
            temp=100-(30+(Level-6)*4-
Timetemp)*100/(30+(Level-6)*4);

            g.drawString("やったー",40,60);
            g.drawString("記録更新:"+temp+"点",25,72);
        }
        else{
            temp=100-Time[Level-6]*100/(30+(Level-6)*4);
            g.drawString("  残念  ",40,60);
            g.drawString("最高記録:"+temp+"点",25,72);
        }
        g.drawString("次へ  ﾘﾄﾗｲ  ﾀｲﾄﾙ",12,96);
        break;
    case 3://ゲームオーバー
        g.setColor(g.getColorOfRGB(255-R,255-G,255-B));
        g.drawString("GAME OVER ",35,70);
        g.setColor(g.getColorOfRGB(0,255,0));
        if(menu!=1)g.fillRect(menu*45,84,24,12);
        else g.fillRect(36,84,44,12);
        g.setColor(g.getColorOfRGB(0,0,255));
        g.drawString("ﾘﾄﾗｲ  NewGame  ﾀｲﾄﾙ",0,96);
        break;
    case 4://タイトルメニュー
        g.setColor(g.getColorOfRGB(0,0,127));
        g.fillRect(34,14,60,105);
        g.setColor(g.getColorOfRGB(0,0,255));
        g.fillRect(30,10,60,105);
        g.setColor(g.getColorOfRGB(255,255,0));
```

★メンバーのみなさまは，いつまでも変わらずお元気でなによりですね。（神奈川県川崎市・林　雄一29歳）……**【Dr.D：ワシはいつでも元気じゃ。まだまだ若いもんには負けない！影：でた！Dr.の口癖。年をとるほど，「若いもんには…」とか言っちゃうんだよね。】**

```
				g.fillRect(35,31+menu*12,50,12);
				g.setColor(g.getColorOfRGB(255,255,0));
				g.drawString("メニュー",35,22);
				g.setColor(g.getColorOfRGB(255,0,0));
				g.drawString("続きから",35,42);
				g.drawString("最初から",35,54);
				g.drawString(" スコア ",35,66);
				g.drawString("ｼｽﾃﾑ設定",35,78);
				g.drawString(" ヘルプ ",30,90);
				g.drawString("  情報  ",35,102);
				g.drawString("  戻る  ",35,114);
				break;
			case 5://ゲーム中メニュー
				g.setColor(g.getColorOfRGB(0,0,127));
				g.fillRect(34,24,60,80);
				g.setColor(g.getColorOfRGB(0,0,255));
				g.fillRect(30,20,60,80);
				g.setColor(g.getColorOfRGB(255,255,0));
				g.fillRect(35,41+menu*12,50,12);
				g.setColor(g.getColorOfRGB(255,255,0));
				g.drawString("メニュー",35,32);
				g.setColor(g.getColorOfRGB(255,0,0));
				g.drawString("リトライ",35,52);
				g.drawString("New Game",35,64);
				g.drawString(" 面ｾﾚｸﾄ ",35,76);
				g.drawString("タイトル",35,88);
				g.drawString("  戻る  ",35,100);
				break;
			case 6://スコア
				Total=0;
				g.setColor(g.getColorOfRGB(255-R,255-G,255-B));
				g.drawString("      スコア      ",0,12-menu);
				g.setColor(g.getColorOfRGB(0,0,255));
				for(i=0,j=0;i<40;i++)
					if(Time[i]!=0){
						temp=100-Time[i]*100/(30+i*4);
						Total+=temp;
						g.drawString(i+"面:"+temp+"点
"+Time[i]+"秒",0,72+j*12-menu);
						j++;
					}
				if(j!=0)temp=Total/j;
				else temp=0;
				g.drawString("トータル:"+Total+"点",0,36-menu);
				g.drawString("平    均:"+temp+"点",0,48-menu);
				break;
			case 7://システム
				g.setColor(g.getColorOfRGB(255,255,0));
				if(menu==0)g.fillRect(0,36,36,12);
				else if(menu==4)g.fillRect(0,84,42,12);
				else g.fillRect(12,36+menu*12,12,12);
				g.setColor(g.getColorOfRGB(255-R,255-G,255-B));
				g.drawString("      システム     ",0,12);
				g.setColor(g.getColorOfRGB(0,0,255));
				g.drawString("背景色:             ",0,48);
				g.drawString("   赤("+R+")          ",0,60);
				g.drawString("   緑("+G+")          ",0,72);
				g.drawString("   青("+B+")          ",0,84);
				if(backL==1){
					g.drawString("ﾊﾞｯｸﾗｲﾄ:<ON> OFF ",0,96);
					PhoneSystem.setAttribute(0,1);
				}
				else{
					g.drawString("ﾊﾞｯｸﾗｲﾄ: ON <OFF>",0,96);
					PhoneSystem.setAttribute(0,0);
				}
				break;
			case 8://ヘルプ
				g.setColor(g.getColorOfRGB(255-R,255-G,255-B));
				if(menu>=0 &&menu< 12)g.drawString("      ヘルプ      ",0,12-menu);
				g.setColor(g.getColorOfRGB(255,255,0));
				if(menu>=0 &&menu< 48)g.drawString("      ルール      ",1,48-menu);
				g.setColor(g.getColorOfRGB(0,0,255));
				if(menu>= 0&&menu< 60)g.drawString("・同じ石をペアで消し",1,60-menu);
				if(menu>= 0&&menu< 72)g.drawString("ていきます。すべての",1,72-menu);
				if(menu>= 0&&menu< 84)g.drawString("石が消えるとクリアで",1,84-menu);
				if(menu>= 0&&menu< 96)g.drawString("す。                ",1,96-menu);
				if(menu>= 0&&menu<108)g.drawString("・石は動き出すと、他",1,108-menu);
				if(menu>= 0&&menu<120)g.drawString("の石に当たるまでとま",1,120-menu);
				if(menu>= 0&&menu<132)g.drawString("りません。同じ石に当",1,132-menu);
				if(menu>= 0&&menu<144)g.drawString("たると、二つとも消え",1,144-menu);
				if(menu> 12&&menu<156)g.drawString("ます。              ",1,156-menu);
				if(menu> 24&&menu<168)g.drawString("・石が画面から飛び出",1,168-menu);
				if(menu> 36&&menu<180)g.drawString("すとミスです。      ",1,180-menu);
				g.setColor(g.getColorOfRGB(255,255,0));
				if(menu> 60&&menu<204)g.drawString("     操作説明     ",1,204-menu);
				g.setColor(g.getColorOfRGB(0,0,255));
				if(menu> 72&&menu<216)g.drawString("  動かしたい石の上に",1,216-menu);
				if(menu> 84&&menu<228)g.drawString("黄色のカーソル(  )",1,228-menu);
				g.setColor(g.getColorOfRGB(255,255,0));
				if(menu> 84&&menu<228)g.drawString("                □ ",1,228-menu);
				g.setColor(g.getColorOfRGB(0,0,255));
				if(menu> 96&&menu<240)g.drawString("をもっていきます。そ",1,240-menu);
				if(menu>108&&menu<252)g.drawString("して決定キーで決定し",1,252-menu);
				if(menu>120&&menu<264)g.drawString("ます。するとカーソル",1,264-menu);
				if(menu>132&&menu<276)g.drawString("が赤色(  )に変わり",1,276-menu);
				g.setColor(g.getColorOfRGB(255,0,0));
				if(menu>132&&menu<276)g.drawString("      □          ",1,276-menu);
				g.setColor(g.getColorOfRGB(0,0,255));
				if(menu>144&&menu<288)g.drawString("ます。そして、動かし",1,288-menu);
				if(menu>156&&menu<300)g.drawString("たい方向のキーを押し",1,300-menu);
				if(menu>168&&menu<312)g.drawString("ます。決定を解除した",1,312-menu);
				if(menu>180&&menu<324)g.drawString("い場合は、もう一度決",1,324-menu);
				if(menu>192&&menu<336)g.drawString("定キーを押すとカーソ",1,336-menu);
				if(menu>204&&menu<348)g.drawString("ルが、黄色に戻ります",1,348-menu);
				if(menu>216&&menu<360)g.drawString("。                  ",1,360-menu);
				if(menu>228&&menu<372)g.drawString("※一部の機種では面の",1,372-menu);
				if(menu>240&&menu<384)g.drawString("作成に大変時間がかか",1,384-menu);
				if(menu>240&&menu<396)g.drawString("ります。            ",1,396-menu);
				break;
			case 9://情報
				g.setColor(g.getColorOfRGB(255-R,255-G,255-B));
				g.drawString("      情報",0,12);
				g.drawString("製作者 :辻  正志",0,48);
				g.drawString("ﾊﾞｰｼﾞｮﾝ: 1.0",0,60);
				break;
			case 10://NowLoading画面
				g.setColor(g.getColorOfRGB(255-R,255-G,255-B));
				g.drawString("    NowCreating     ",0,36);
				g.setColor(g.getColorOfRGB(255-R,255-G,255-B));
				g.drawString("♪ただいま面作成中♪",0,64);
				temp=load*100/(Level+30);
				g.drawString("      "+temp+"%完了",0,76);
				g.setColor(g.getColorOfRGB(0,0,255));
				g.fillRect(9,76,102,12);
				g.setColor(g.getColorOfRGB(255,255,0));
				g.fillRect(10,77,temp,10);
				g.setColor(g.getColorOfRGB(R,G,B));
				g.fillRect(10+temp,77,100-temp,10);
				if(temp>=100)Game=1;
				break;
```

★1Gクロック所有の初心者を教える私のマシンは，ソケット7の450Mクロックです。マシンパワーが力ではないですねぇ。(愛知県知多郡・四谷一獲26歳) ……【影：Dr.なんて，今でも日立の「ベーシックマスターレベル3」に愛着を持ってますしね。Dr.D：うっ。】

```
			}
			g.unlock(false);//ロック解除
		}
	}

	//■■■■■■イベント関数■■■■■■■
	public void processEvent(int msg, int param){
		switch(msg){
		case Display.KEY_PRESSED_EVENT://キーイベント
			switch(Game){
			case 0://タイトル
				if(param==Display.KEY_SOFT1){
					menu=6;Game=4;
				}
				break;
			case 1://ゲーム中
				if(Move==0){
					switch(param){
					case Display.KEY_6:
					case Display.KEY_RIGHT:
						if(cc==1){Move=1;mh=1;}
						else if(++cx==11)cx=1;
						break;
					case Display.KEY_4:
					case Display.KEY_LEFT:
						if(cc==1){Move=1;mh=3;}
						else if(--cx==0)cx=10;
						break;
					case Display.KEY_8:
					case Display.KEY_DOWN:
						if(cc==1){Move=1;mh=2;}
						else if(++cy==11)cy=1;
						break;
					case Display.KEY_2:
					case Display.KEY_UP:
						if(cc==1){Move=1;mh=0;}
						else if(--cy==0)cy=10;
						break;
					case Display.KEY_5:
					case Display.KEY_SELECT:
						if(cc==1)cc=0;
						else if(Map[cy][cx]!=0)cc=1;
						break;
					case Display.KEY_SOFT1:
						menu=4;temp=0;Game=5;break;
					}
					mx=cx;my=cy;
				}
				break;
			case 2://クリア
				switch(param){
				case Display.KEY_6:
				case Display.KEY_RIGHT:
					menu++;
					if(menu==3)menu=0;
					break;
				case Display.KEY_4:
				case Display.KEY_LEFT:
					menu--;
					if(menu==-1)menu=2;
					break;
				case Display.KEY_5:
				case Display.KEY_SELECT:
				if(Time[Level-6]>30+(Level-6)*4-Timetemp||Time[Level-
6]==0)
					Time[Level-6]=30+(Level-6)*4-Timetemp;
					if(Level==HiLevel)HiLevel++;
					switch(menu){
					case 0:Level++;NewGame();break;
					case 1:ReTry();break;
					case 2:Game=0;break;
					}
					break;
				}
				break;
			case 3://ゲームオーバー
				switch(param){
				case Display.KEY_6:
				case Display.KEY_RIGHT:
					menu++;
					if(menu==3)menu=0;
					break;
				case Display.KEY_4:
				case Display.KEY_LEFT:
					menu--;
					if(menu==-1)menu=2;
					break;
				case Display.KEY_5:
				case Display.KEY_SELECT:
					switch(menu){
					case 0:ReTry();break;
					case 1:NewGame();break;
					case 2:Game=0;break;
					}
					break;
				}
				break;
			case 4://タイトルメニュー
				switch(param){
				case Display.KEY_8:
				case Display.KEY_DOWN:
					menu++;
					if(menu==7)menu=0;
					break;
				case Display.KEY_2:
				case Display.KEY_UP:
					menu--;
					if(menu==-1)menu=6;
					break;
				case Display.KEY_5:
				case Display.KEY_SELECT:
					switch(menu){
					case 0:Level=HiLevel;NewGame();break;
					case 1:Dlg1.setText("セーブデータが消えます。
よろしいですか?");
						if(Dlg1.show()==Dlg1.BUTTON_YES){
Level=HiLevel=6;for(i=0;i<40;i++)Time[i]=0;
							NewGame();
						}
						else Game=0;
						break;
					case 2:Game=6;menu=0;break;
					case 3:Game=7;menu=0;break;
					case 4:Game=8;menu=0;break;
					case 5:Game=9;break;
					case 6:Game=0;break;
					}
					break;
				}
				break;
			case 5://ゲーム中メニュー
```

★2月号の「チャレンジ！オリジナル言語制作への道」を読んで，自分もオリジナル言語を作ってみたくなりました。ベースはVB5で。
(青森県南津軽郡・ていこく22歳) ……【影：興味がそそられたのなら，うれしいぞ。読者の反応がないものは寂しいものがある。】

```
				switch(param){
				case Display.KEY_6:
				case Display.KEY_RIGHT:
					if(menu==2){
						temp++;
if(Level+temp>HiLevel)temp=HiLevel-Level;
					}
					break;
				case Display.KEY_4:
				case Display.KEY_LEFT:
					if(menu==2){
						temp--;
						if(Level+temp==5)temp=6-Level;
					}
					break;
				case Display.KEY_8:
				case Display.KEY_DOWN:
					menu++;
					if(menu==5)menu=0;
					break;
				case Display.KEY_2:
				case Display.KEY_UP:
					menu--;
					if(menu==-1)menu=4;
					break;
				case Display.KEY_5:
				case Display.KEY_SELECT:
					switch(menu){
					case 0:ReTry();break;
					case 1:NewGame();break;
					case 2:Level+=temp;NewGame();break;
					case 3:Game=0;break;
					case 4:Game=1;break;
					}
					break;
				}
				break;
			case 6://スコア
				switch(param){
				case Display.KEY_8:
				case Display.KEY_DOWN:
					menu+=12;
					if(menu>256)menu=256;
					break;
				case Display.KEY_2:
				case Display.KEY_UP:
					menu-=12;
					if(menu<0)menu=0;
					break;
				case Display.KEY_5:
				case Display.KEY_SELECT:Game=0;break;
				}
				break;
			case 7://システム
				switch(param){
				case Display.KEY_6:
				case Display.KEY_RIGHT:
					switch(menu){
					case 1:R++;if(R==256)R=0;break;
					case 2:G++;if(G==256)G=0;break;
					case 3:B++;if(B==256)B=0;break;
					case 4:backL=0;break;
					}
					break;
				case Display.KEY_4:
				case Display.KEY_LEFT:
					switch(menu){
					case 1:R--;if(R==-1)R=255;break;
					case 2:G--;if(G==-1)G=255;break;
					case 3:B--;if(B==-1)B=255;break;
					case 4:backL=1;break;
					}
					break;
				case Display.KEY_8:
				case Display.KEY_DOWN:
					menu++;
					if(menu==5)menu=0;
					break;
				case Display.KEY_2:
				case Display.KEY_UP:
					menu--;
					if(menu==-1)menu=4;
					break;
				case Display.KEY_5:
				case Display.KEY_SELECT:Game=0;break;
				}
				break;
			case 8://ヘルプ
				switch(param){
				case Display.KEY_8:
				case Display.KEY_DOWN:
					menu+=12;
					if(menu>268)menu=268;
					break;
				case Display.KEY_2:
				case Display.KEY_UP:
					menu-=12;
					if(menu<0)menu=0;
					break;
				case Display.KEY_5:
				case Display.KEY_SELECT:Game=0;break;
				}
				break;
			case 9://情報
				if(param==Display.KEY_SELECT||param==Display.KEY_5)
					Game=0;
				break;
			}
			break;
		case Display.TIMER_EXPIRED_EVENT://タイマーイベント
			timer++;//1秒毎にカウント
			break;
		}
	}
}
```

《リスト3》iEgypt.jamのリスト

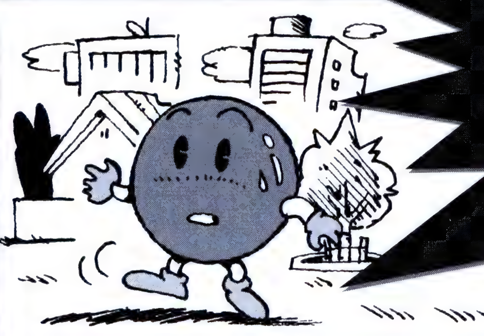
```
AppName = iエジプトの石
PackageURL = iEgypt.jar
AppSize = 9767
AppClass = iEgypt
LastModified = Fri, 16 Nov 2001 03:16:14
AppVer = 1.0
SPsize = 500
```

★毎年恒例…年賀ハガキが余った。久々に大量投稿か？笑い。（茨城県古河市・**年齢不相応41歳**）……【**つぐ美**：というわけで，うわーっ！たくさんの年賀ハガキが届きました。**編**：これ全部載せてたら，何年持つことか。**つぐ美**：お年玉当たりハガキ見一つけた。】

iモード対応携帯電話（iアプリ）用

慣性移動を克服しろ

ころころやかもてぃ



Mr．親父（19歳）

ゲーム内容

「やかもてい」とは，自分と友人との間で自然発生したキャラクターです。設定としては「魔法を主食とする亜動物」です。マスコットキャラとしてよさそうなので使いました。

「やかもてぃ」を,画面中央に表示される四角の枠まで移動させるゲームです。

操作方法

携帯電話の番号ボタンで「やかもてい」を操作します。

ボタンの対応は以下のようになります。2：上移動，4：左移動，6：右移動，8：下移動，5：決定，ソフトキー1：リセット，ソフトキー2：終了。

プログラムの説明

このゲームを作るにあたってキャラクタの移動処理で慣性をぜひ実現させたかったため，「大雑把であるが，そう見えなくも無い」というぐらいの慣性をつけました。

temp_move_x，temp_move_yを加算，減算させ，その値が「最大移動量」，「0」になったらmove_x,move_yを加算，減算させるというものです。

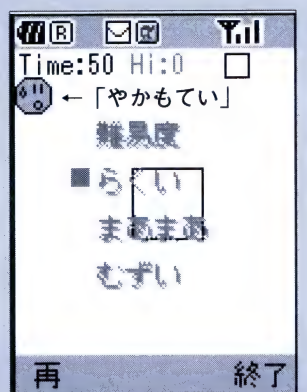
実機とパソコンで実行するときの最も大きな相違は「速度」だと思います。そこで「始めの数秒を使い，実行環境の描画速度を調べ，それで調整を行う」という方法を導入しました。

当初，「やかもてぃ」のグラフィックデータは，2色gifファイルを使っていましたが，容量がどうしても肥大化してしまうため，データテーブル化することに変更しました。

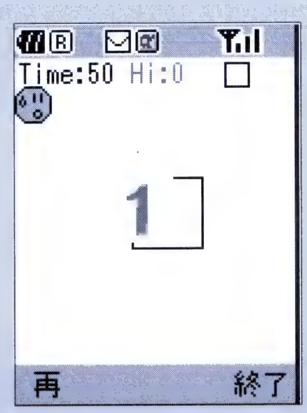
いろいろ

これまではi-jadeでデバッグしただけで，実機では動かしてませんでしたが，友人にi-modeを借りて実際に動作を確認しました。

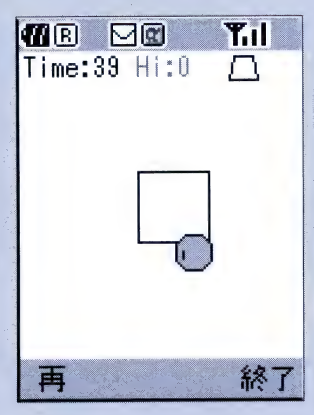
実際にお盆の上でボールを動かした感覚を，そのまま「やかもてぃ」の動きにしたら，ゲームの難易度が「大甘」になりました。慣れてしまえばすぐにハイスコアが出せると思います。



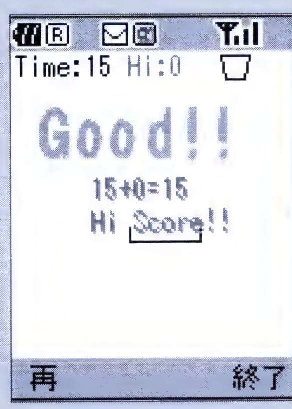
《写真1》オリジナルキャラ「やかもてぃ」を操作



《写真2》カウントダウン後にゲームスタート



《写真3》慣性のついた動きが特徴



《写真4》四角の中に移動させればクリアとなる

★Kylixを取り上げてください。（大阪市・玉出のタマ36歳）……【編：熱心な読者です。毎月，同じ文面で投稿してきます。Dr.，読者の要望になんとか応えてくださいよ。Dr.D：Linuxでのビジュアル開発に優れたプログラミング・ツールじゃな。】

《リスト１》koroyaka.javaのリスト

```
// ころころやかもてぃ
// koroyaka.java

import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import java.io.*;
import java.util.*;
import javax.microedition.io.*;

public class koroyaka extends IApplication{
        public void start(){
                Display.setCurrent(new GameScreen());
        }
}

class GameScreen extends Canvas implements Runnable{
        Thread myThread;

        int num[] = new int[2];                                         // イメージ番号
        int yaka_x = 0;                                                 // やかもてぃのx
座標
        int yaka_y = 11;                                        // やかもてぃのy座標
        int inKey[] = new int[2];                               // キー入力の状態
        int move_x = 0,move_y = 0;                              // x,y方向の変化量
        int temp_move_x = 4,temp_move_y = 4;                    // x,y方向の最大変化量
        int move_swich_x = 0,move_swich_y = 0;                  // x,y方向移動の状態flag
        int start = 0,end = 0;                                  // 開始，終了flag
        int level = 0;                                                  // 難易度設定
        int time = 54,hitime = 0,second = 0,temp_second = 0;    // 時間計測用
        int wait = 20,temp_wait = 0;                            // 速度設定用

        // やかもてぃ顔データ
        int [][][] yaka_face = {
{{ 16*3+9, 16*3+12, 16*4+4, 16*4+9, 16*4+12, 16*5+4, 16*5+9, 16*5+12, 16*6+3, 16*6+5, 16*6+9, 16*6+12,
16*7+3, 16*7+5, 16*8+4, 16*10+10, 16*10+11, 16*11+9, 16*11+12, 16*12+9, 16*12+12, 16*13+10, 16*13+11},
 { 16*3+5, 16*3+8, 16*4+5, 16*4+8, 16*5+5, 16*5+8, 16*6+5, 16*6+8, 16*10+6, 16*10+7, 16*11+5, 16*11+8,
16*12+5, 16*12+8, 16*13+6, 16*13+7, 0, 0, 0, 0, 0, 0, 0},
 { 16*3+4, 16*4+4, 16*5+4, 16*6+4, 16*10+2, 16*10+3, 16*11+4, 16*12+4, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 { 16*6+15, 16*7+15, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 { 16*4+12, 16*5+12, 16*6+11, 16*6+13, 16*7+11, 16*7+13, 16*8+12, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 { 16*3+13, 16*4+8, 16*4+13, 16*5+8, 16*5+13, 16*6+7, 16*6+9, 16*6+13, 16*7+7, 16*7+9,
16*8+8, 16*10+14, 16*10+15, 16*11+13, 16*12+13, 0, 0, 0, 0, 0, 0, 0,
0}},
{{ 16*7+9, 16*7+12, 16*8+4, 16*8+9, 16*8+12, 16*9+4, 16*9+9, 16*9+12, 16*10+3, 16*10+5, 16*10+9, 16*10+12,
16*11+3, 16*11+5, 16*12+4, 16*14+10, 16*14+11, 0, 0, 0, 0, 0, 0},
 { 16*7+5, 16*7+8, 16*8+5, 16*8+8, 16*9+5, 16*9+8, 16*10+5, 16*10+8, 16*14+6, 16*14+7, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 { 16*7+4, 16*8+4, 16*9+4, 16*10+4, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {16*10+15, 16*11+15, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 { 16*8+12, 16*9+12, 16*10+11, 16*10+13, 16*11+11, 16*11+13, 16*12+12, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 { 16*7+13, 16*8+8, 16*8+13, 16*9+8, 16*9+13, 16*10+7, 16*10+9, 16*10+13, 16*11+7, 16*11+9, 16*12+8, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}},
{{ 16*11+9, 16*11+12, 16*12+4, 16*12+9, 16*12+12, 16*13+4, 16*13+9, 16*13+12, 16*10+3, 16*14+5, 16*14+9, 16*14+12,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 { 16*11+5, 16*11+8, 16*12+5, 16*12+8, 16*13+5, 16*13+8, 16*14+5, 16*14+8, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 { 16*11+4, 16*12+4, 16*13+4, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {16*12+12, 16*13+12, 16*14+11, 16*14+13, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
 {16*11+13, 16*12+8, 16*12+13, 16*13+8, 16*13+13, 16*14+7, 16*14+9, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}},
{{ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
```


入力方法は33ページをご覧ください。

使用している画像ファイル

1 2 3 GO!!
Good!! Over Time

上段左から，1.gif，2.gif，3.gif，go.gif
下段左から，good.gif，over.gif，time.gif

CHECKER FLAG

編：手軽に短時間で遊べるゲームなので，チョットした暇つぶしで遊ぶのに，ちょうど良いゲームかもしれません。ただ，キー割り当てが人によっては違和感を覚えるかも？　やはりカーソルキーでも遊べるようにしてほしかったですね。

つぐ美：ただ，内容のわりにリストが長いわねぇ〜。まぁ，リストの３分の１をグラフィックデータが占めているのを考えると，しょうがないかもしれないわね。

Dr.D：あとは，動きの処理を行う部分の記述に無駄がある感じじゃな。X方向，Y方向と違う変数を用いるため，似たような処理をそれぞれ別々に処理しているみたいじゃが，やはり配列を用いたり，関数(メソッド)で引数渡しにするなどの工夫はできるのではないかな？

《第1表》変数表

num[]	やかもてぃ顔データの配列番号 num[0]がyデータ，num[1]がx方向
yaka_x	やかもてぃのx座標
yaka_y	やかもてぃのy座標
inKey[]	キー入力の状態 inKey[0]がy方向，num[1]がx方向
move_x	x方向の移動量
move_y	y方向の移動量
temp_move_x	x方向の最大変化量(慣性表現のために加減させる)
temp_move_y	y方向の最大移動量(慣性表現のために加減させる)
move_swich_x	x方向移動の状態flag(左移動中か右移動中か)
move_swich_y	y方向移動の状態flag(上移動中か下移動中か)
start	開始flag
end	終了flag
level	難易度設定
time	現時間
hitime	ハイスコア
second	一秒間隔を調べるために使用
temp_second	secondデータの一時保管用
wait	速度設定(ゲーム中で自動更新)
temp_wait	wait生成のために使用
iKey	キー入力
yaka_face[][][]	やかもてぃグラフィックデータ
one	イメージ(1)
two	イメージ(2)
three	イメージ(3)
go	イメージ(GO!!)
good	イメージ(Good)
time_	イメージ(Time)
over	イメージ(Over)

★きょーこさんはいつになったら帰ってくるのでしょう…。（福島県郡山市・**近内沖尋19歳**）……【**つぐ美**：きょうこさん？なによ，私がいるっていうのに！**影**：知ーらない，つぐ美ちゃんを怒らせたら大変だぞ。**つぐ美**：影さん，やけ食いに付き合って！。**影**：うっ。】

```
  {       0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  {       0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  {       0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  {       0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  {       0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0}},
{{ 16*2+10, 16*2+11, 16*3+9, 16*3+12, 16*4+9, 16*4+12, 16*5+10, 16*5+11,  0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  {  16*2+6, 16*2+7, 16*3+5, 16*3+8, 16*4+5, 16*4+7, 16*5+6, 16*5+7,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  {  16*3+4, 16*4+4, 16*5+2, 16*5+3,      0,      0,      0,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  {       0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  {       0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  { 16*3+13, 16*4+13, 16*5+14, 16*5+15,   0,      0,      0,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0}},
{{  16*1+9, 16*1+12, 16*2+5, 16*2+9, 16*2+12, 16*3+3, 16*3+5, 16*4+4, 16*6+10, 16*6+11, 16*7+9,
16*7+12,  16*8+9, 16*8+12, 16*9+10, 16*9+11,      0,      0,      0,      0,      0,      0,
0},
  {  16*1+5, 16*1+8, 16*2+5, 16*2+8, 16*6+6, 16*6+7, 16*7+5, 16*7+8, 16*8+5, 16*8+8, 16*9+6,
16*9+7,       0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0},
  {  16*2+4, 16*6+2, 16*6+2, 16*7+4, 16*8+4, 16*9+2, 16*9+3,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  {       0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  { 16*1+12, 16*2+11, 16*2+13, 16*3+11, 16*3+13, 16*4+12,   0,      0,      0,      0,      0,
0,        0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0},
  {  16*1+8, 16*2+7, 16*2+9, 16*2+13, 16*3+7, 16*3+9, 16*4+8, 16*6+14, 16*6+15, 16*7+13, 16*8+13,
16*9+14, 16*9+15,     0,      0,      0,      0,      0,      0,      0,      0,      0,
0}}};

        Image one,two,three,go,good,time_,over;                     // 演出用gif

        public Image getImg(String sName){          // イメージ取得
                MediaImage mi = MediaManager.getImage(sName);
                try{
                        mi.use();
                }catch(ConnectionException e){}
                return(mi.getImage());
        }

        public GameScreen(){                        // コントラクタ
                // 画像読み込み
                one = getImg("resource:///1.gif");
                two = getImg("resource:///2.gif");
                three = getImg("resource:///3.gif");
                go = getImg("resource:///go.gif");
                good = getImg("resource:///good.gif");
                time_ = getImg("resource:///time.gif");
                over = getImg("resource:///over.gif");

                try{
                        DataInputStream in = Connector.openDataInputStream("scratchpad:///0");
                        hittime = in.readInt();
                        in.close();
                }catch(IOException e){}

                myThread = new Thread(this);                // スレッド作成
                myThread.start();                           // スレッド開始
        }

        public void paint(Graphics g){                      // 描画メソッド
                int i;
                int temp_num[] = new int[2];
                int width = getWidth();
                int height = getHeight();
```

```
                temp_num[0] = num[0] / 4;
                temp_num[1] = num[1] / 4;

                Calendar cal = Calendar.getInstance();          // 時間計測
                second = cal.get(Calendar.SECOND);
                if (temp_second != second && start == 1 && end != 1){
                        temp_second = second;
                        if (time > 0){
                                time--;
                        }
                }

                // 慣性表現
                if(inKey[0] == -1){
                        if (move_swich_y != -1){
                                move_swich_y = 1;
                                temp_move_y -= move_y;
                                if (temp_move_y < 1 && move_y < 4){
                                        move_y++;
                                        temp_move_y = 4;
                                }
                                yaka_y += move_y;
                        }
                        else{
                                temp_move_y -= move_y;
                                if (temp_move_y < 1 && move_y > 0){
                                        move_y--;
                                        temp_move_y = 4;
                                }
                                yaka_y -= move_y;
                                if (move_y == 0) {
                                        move_y = 1;
                                        move_swich_y = 1;
                                }
                        }

                        num[0] += move_y;
                        if (num[0] > 23)
                        {
                                num[0] = 0;
                        }
                }
                else if(inKey[0] == 1){
                        if (move_swich_y != 1){
                                move_swich_y = -1;
                                temp_move_y -= move_y;
                                if (temp_move_y < 1 && move_y < 4){
                                        move_y++;
                                        temp_move_y = 4;
                                }
                                yaka_y -= move_y;
                        }
                        else{
                                temp_move_y -= move_y;
                                if (temp_move_y < 1 && move_y > 0){
                                        move_y--;
                                        temp_move_y = 4;
                                }
                                yaka_y += move_y;
                                if (move_y == 0) {
                                        move_y = 1;
                                        move_swich_y = -1;
                                }
                        }
                        num[0] -= move_y;
                        if (num[0] < 0)
                        {
                                num[0] = 23;
                        }
                }
                else if(inKey[0] == 0){
                        if (move_swich_y == -1){
                                temp_move_y -= move_y;
```


★HSPやっているので，その記事が出ていてうれしいです。（東京都日野市・**小島出雲15歳**）……【**Dr.D**：HSPは「オンラインソフトウェア大賞2001」で入賞するなど，人気のフリーソフトじゃ。結構，学校でも使われているようじゃぞ。ベーマガでも連載中じゃ。】

```
			if (temp_move_y < 1 && move_y > 0){
				move_y--;
				temp_move_y = 8;
			}
			if (move_y > 0){
				yaka_y -= move_y;
			}
			else{
				move_swich_y = 0;
			}

			num[0] += move_y;
			if (num[0] > 23)
			{
				num[0] = 0;
			}
		}
		else if (move_swich_y == 1){
			temp_move_y -= move_y;
			if (temp_move_y < 1 && move_y > 0){
				move_y--;
				temp_move_y = 8;
			}
			if (move_y > 0){
				yaka_y += move_y;
			}
			else{
				move_swich_y = 0;
			}
			num[0] -= move_y;
			if (num[0] < 0)
			{
				num[0] = 23;
			}
		}
	}

	if(inKey[1] == -1){
		if (move_swich_x != 1){
			move_swich_x = -1;
			temp_move_x -= move_x;
			if (temp_move_x < 1 && move_x < 4){
				move_x++;
				temp_move_x = 4;
			}
			yaka_x -= move_x;
		}
		else{
			temp_move_x -= move_x;
			if (temp_move_x < 1 && move_x > 0){
				move_x--;
				temp_move_x = 4;
			}
			yaka_x += move_x;
			if (move_x == 0){
				move_x = 1;
				move_swich_x = -1;
			}
		}

		num[1] += move_x;
		if (num[1] > 23){
			num[1] = 0;
		}
	}
	else if(inKey[1] == 1){
		if (move_swich_x != -1){
			move_swich_x = 1;
			temp_move_x -= move_x;
			if (temp_move_x < 1 && move_x < 4){
				move_x++;
				temp_move_x = 4;
			}
			yaka_x += move_x;
```

```
		}
		else{
			temp_move_x -= move_x;
			if (temp_move_x < 1 && move_x > 0){
				move_x--;
				temp_move_x = 4;
			}
			yaka_x -= move_x;
			if (move_x == 0){
				move_x = 1;
				move_swich_x = 1;
			}
		}

		num[1] -= move_x;
		if (num[1] < 0){
			num[1] = 23;
		}
	}
	else if(inKey[1] == 0){
		if (move_swich_x == 1){
			temp_move_x -= move_x;
			if (temp_move_x < 1 && move_x > 0){
				move_x--;
				temp_move_x = 8;
			}
			if (move_x > 0){
				yaka_x += move_x;
			}
			else{
				move_swich_x = 0;
			}
			num[1] += move_x;
			if (num[1] > 23){
				num[1] = 0;
			}
		}
		else if (move_swich_x == -1){
			temp_move_x -= move_x;
			if (temp_move_x < 1 && move_x > 0){
				move_x--;
				temp_move_x = 8;
			}
			if (move_x > 0){
				yaka_x -= move_x;
			}
			else{
				move_swich_x = 0;
			}
			num[1] -= move_x;
			if (num[1] < 0){
				num[1] = 23;
			}
		}
	}

	// 移動領域指定
	if (yaka_x < 1){
		yaka_x = 0;
	}
	if (yaka_x > width - 16 - 1){
		yaka_x = width -16;
	}
	if (yaka_y < 12){
		yaka_y = 11;
	}
	if (yaka_y > height - 16 - 1){
		yaka_y = height -16;
	}

	// 速度調整
	if (start == 1){
		if (time >= 52){
			temp_wait++;
```

iアプリ

```
                }
                else{
                        wait = temp_wait;
                }
        }

        // 描画
        g.lock();
                g.setColor(g.getColorOfRGB(255, 255, 255));
                g.fillRect(0, 0, getWidth(), getHeight());
                g.setColor(g.getColorOfRGB(255, 0, 0));
                g.drawString("Hi:" + Integer.toString(hitime),50,10);    // ハイスコア
                g.setColor(g.getColorOfRGB(0, 0, 0));
                if (time <= 50 && time >= 0){
                        g.drawString("Time:" + Integer.toString(time),2,10);    //
時間
                }
                else if(time > 50){
                        g.drawString("Time:50",2,10);
                }
                else{
                        g.drawString("Time:0",2,10);
                }
                g.drawLine(90-(move_y * move_swich_y / 3),1-(move_x * move_swich_x /
3),100+(move_y * move_swich_y / 3),1+(move_x * move_swich_x / 3));          // 現在の傾きの
状態を表示
                g.drawLine(90+(move_y * move_swich_y / 3),1-(move_x * move_swich_x / 3)
+ 10,100-(move_y * move_swich_y / 3),1+(move_x * move_swich_x / 3) + 10);    //
                g.drawLine(90-(move_y * move_swich_y / 3),1-(move_x * move_swich_x /
3),90+(move_y * move_swich_y / 3),1-(move_x * move_swich_x / 3) + 10);       //
                g.drawLine(100+(move_y * move_swich_y / 3),1+(move_x * move_swich_x /
3),100-(move_y * move_swich_y / 3),1+(move_x * move_swich_x / 3) + 10);      //
                g.fillRect(50 + level , 50 + level, 31 - level, 31 - level);
                g.setColor(g.getColorOfRGB(255, 255, 255));
                g.fillRect(51 + level, 51 + level, 29 - level, 29 - level);

                g.setColor(g.getColorOfRGB(192, 192, 192));  // やかもてい中身
                for (i = 1;i <= 3; i++){
                        g.drawLine(yaka_x + i,yaka_y + 5 - i,yaka_x + i,yaka_y + 10 +
i);
                        g.drawLine(yaka_x + 15 - i,yaka_y + 5 - i,yaka_x + 15 -
i,yaka_y + 10 + i);
                }
                g.fillRect(yaka_x + 4,yaka_y + 1, 8, 14);

                g.setColor(g.getColorOfRGB(0, 0, 0));                // やかもてい輪
郭
                g.drawLine(yaka_x + 4,yaka_y,yaka_x + 11,yaka_y);
                g.drawLine(yaka_x + 4,yaka_y + 15,yaka_x + 11,yaka_y + 15);
                g.drawLine(yaka_x + 11,yaka_y,yaka_x + 15,yaka_y + 4);
                g.drawLine(yaka_x + 15,yaka_y + 4,yaka_x + 15,yaka_y + 11);
                g.drawLine(yaka_x + 15,yaka_y + 11,yaka_x + 11,yaka_y + 15);
                g.drawLine(yaka_x + 4,yaka_y + 15,yaka_x,yaka_y + 11);
                g.drawLine(yaka_x,yaka_y + 11,yaka_x,yaka_y + 4);
                g.drawLine(yaka_x,yaka_y + 4,yaka_x + 4,yaka_y);

                // やかもてい 顔
                for (i=0;i<23;i++){
                        if (yaka_face[temp_num[0]][temp_num[1]][i] != 0){
                                g.drawLine(yaka_x +
((yaka_face[temp_num[0]][temp_num[1]][i] - ((yaka_face[temp_num[0]][temp_num[1]][i] / 16) * 16)) -
1),yaka_y + (yaka_face[temp_num[0]][temp_num[1]][i] / 16),yaka_x +
((yaka_face[temp_num[0]][temp_num[1]][i] - ((yaka_face[temp_num[0]][temp_num[1]][i] / 16) * 16)) -
1),yaka_y + (yaka_face[temp_num[0]][temp_num[1]][i] / 16));
                        }
                }

                if (yaka_x > (50 + level) && yaka_x < (50 + level + (15 - level)) &&
yaka_y > (50 + level) && yaka_y < (50 + level + (15 - level))){
                        g.setColor(g.getColorOfRGB(255, 255, 255));
                        g.fillRect(33,48,50,26);
                        g.drawImage(good, 10 ,20);

                        g.setColor(g.getColorOfRGB(255,127,000));
                        g.drawString(Integer.toString(time) + "+" +
Integer.toString(level * 2) + "=" + Integer.toString(time + level * 2),35,63);
                        g.setColor(g.getColorOfRGB(255, 000, 000));
                        g.drawString(Integer.toString(time) + "+" +
Integer.toString(level * 2) + "=" + Integer.toString(time + level * 2),34,62);

                        if((time + level * 2) >= hitime){
                                g.setFont(Font.getFont(Font.FACE_PROPORTIONAL |
Font.SIZE_LARGE | Font.STYLE_BOLD));

                                g.setColor(g.getColorOfRGB(255,127,000));
                                g.drawString("Hi Score!!",35,77);
                                g.setColor(g.getColorOfRGB(255, 000, 000));
                                g.drawString("Hi Score!!",34,76);
                        }
                        end = 1;
                        inKey[0] = 0;
                        inKey[1] = 0;
                        move_x = 0;
                        move_y = 0;
                }
                else if (time < 1){
                        g.setColor(g.getColorOfRGB(255, 255, 255));
                        g.fillRect(33,48,50,26);
                        g.drawImage(time_, 10 ,30);
                        g.drawImage(over, 50 ,60);

                        end = 1;
                        inKey[0] = 0;
                        inKey[1] = 0;
                        move_x = 0;
                        move_y = 0;
                }
                else if(start == 1){
                        switch(time - 50)
                        {
                                case 1:
                                        g.drawImage(one, 40 ,50);
                                        break;
                                case 2:
                                        g.drawImage(two, 40 ,50);
                                        break;
                                case 3:
                                        g.drawImage(three, 40 , 50);
                                        break;
                                case 0:
                                        g.drawImage(go, 30 ,50);
                                        break;
                        }
                }
                else if(start != 1){
                        g.setFont(Font.getFont(Font.FACE_PROPORTIONAL |
Font.SIZE_LARGE | Font.STYLE_BOLD));

                        g.setColor(g.getColorOfRGB(255,127,000));
                        g.drawString("難易度",36,41);
                        g.drawString("らくい",36,61);
                        g.drawString("まあまあ",36,81);
                        g.drawString("むずい",36,101);
                        g.setColor(g.getColorOfRGB(255,000,000));
                        g.drawString("難易度",35,40);
                        g.drawString("らくい",35,60);
                        g.drawString("まあまあ",35,80);
                        g.drawString("むずい",35,100);
                        g.fillRect(24, 50 + (level / 5 * 21), 8, 8);
                }
        g.unlock(false);
    }

    public void run(){                                  // スレッド実行

        int iKey;
        int key_flag = 0;

        setSoftLabel(Frame.SOFT_KEY_1,"再");
        setSoftLabel(Frame.SOFT_KEY_2,"終了");
```

★内容が充実しすぎていて，1か月じゃ読みきれない。というわけで，たまに買い逃すこともあるんだよなー。（埼玉県所沢市・藤16歳）……【編：それは大変，1か月で読み切れる内容にしなきゃいけないってことか？影さんの仕事がなくなっちゃうよ。影：ゲ！】

```
	for(;;){

		try
		{
			Thread.sleep(wait);	// ウェイト
		}catch(InterruptedException ignore){}

		iKey = getKeypadState();	// キー入力

		if (iKey == (1 << Display.KEY_SOFT1))
		{
			// 再実行
			// 変数初期化
			inKey[0] = 0;
			inKey[1] = 0;
			move_x = 0;
			move_y = 0;
			yaka_x = 0;
			yaka_y = 11;
			num[0] = 0;
			num[1] = 0;
			move_swich_x = 0;
			move_swich_y = 0;
			temp_move_x = 4;
			temp_move_y = 4;
			start = 0;
			end = 0;
			level = 0;
			time = 54;
			second = 0;
			temp_second = 0;
		}

		if (iKey == (1 << Display.KEY_SOFT2))
		{
		// 終了
			IApplication.getCurrentApp().terminate();
		}

		if (end != 1){
			if (start == 1 && time <= 50){
				if (iKey == (1 << Display.KEY_2))
				{
					// UP
					if (move_y == 0){
						move_y = 1;
					}
					inKey[0] = 1;
					inKey[1] = 0;
				}
				if (iKey == (1 << Display.KEY_8))
				{
					// DOWN
					if (move_y == 0){
						move_y = 1;
					}
					inKey[0] = -1;
					inKey[1] = 0;
				}
				if (iKey == (1 << Display.KEY_4))
				{
					// LEFT
					if (move_x == 0){
						move_x = 1;
					}
					inKey[0] = 0;
					inKey[1] = -1;
				}
				if (iKey == (1 << Display.KEY_6))
				{
					// RIGHT
					if (move_x == 0){
						move_x = 1;
					}
					inKey[0] = 0;
					inKey[1] = 1;
				}
			}
			else if(start != 1){
				if(key_flag != 1){
					if (iKey == (1 << Display.KEY_2))
					{
						// UP
						if(level == 5){
							level = 0;
						}
						else if(level == 10){
							level = 5;
						}
						key_flag = 1;
					}
					else if (iKey == (1 << Display.KEY_8))
					{
						// DOWN
						if(level == 0){
							level = 5;
						}
						else if(level == 5){
							level = 10;
						}
						key_flag = 1;
					}
				}
				if (iKey != (1 << Display.KEY_2) && iKey != (1 <<
Display.KEY_8)){

					key_flag = 0;
				}

				if (iKey == (1 << Display.KEY_5))
				{
				// 難易度選択,開始
					start = 1;
				}
			}

			repaint();			// 再描画
		}

		if(end == 1){
			if((time + level * 2) >= hitime)
			{
				try{
					DataOutputStream out =
Connector.openDataOutputStream("scratchpad:///0");

					out.writeInt(time);
					out.close();
					hitime = time + level * 2;
				}catch(IOException e) {}
			}
		}
	}
}
}
```

《リスト2》koroyaka.jamのリスト

```
AppName = koroyaka
PackageURL = koroyaka.jar
AppSize = 7323
AppClass = koroyaka
LastModified = Wed, 28 Nov 2001 21:00:24
SPsize = 8
```

★1月号に引き続き，2月号でもCPUの話，すごく良かったです。これは…連載になるのですか？また，アップグレード講座もありがたい。WindowsXPが出たばっかりだから，参考になります。（神奈川県川崎市・**桜井道孝21歳**）……【**編**：CPUも組み立ても連載です。】

PC-9801シリーズ互換機(N88-日本語BASIC(86)Disk-BASIC)用



タイピングで風船を割っていく

バルーンタイプ

奥田　茂延（25歳）

ゲーム内容

このゲームは今流行(もう終わったのかもしれない)のタイピングゲームです。でも普通のタイプしていくゲームでは面白くないので，鳥を操作するタイピングゲームにしました。

遊びかた

タイピングゲームですので，プレイするにはやはりキーボードを使用します。プログラムをRUNすると，まずキャラクター画像の準備を始めます。

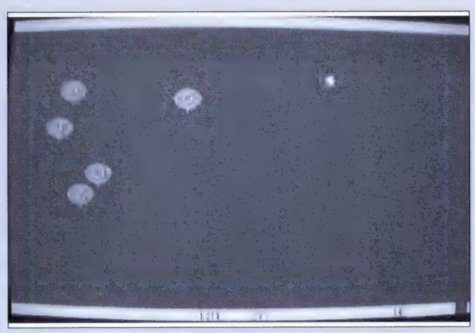
ゲームが始まると，最初に中央に鳥が羽ばたいています。この鳥を操作します。画面下から英字１文字が描いてある風船が飛んで来ます（**写真２**）。この風船に描かれている文字と同じキーを押してください。

すると，鳥は押されたキーと同じ文字の風船に向かって飛んでいきます（**写真３**）。鳥の移動中に，他の風船の文字を打つと目標を変えて飛んでいきます。

ミスタイプが10回になると強制ゲームオーバーです。



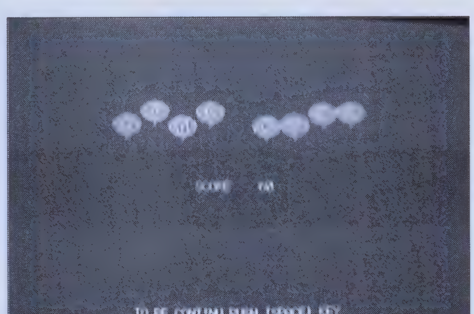
《写真１》風船と鳥が登場するタイピングゲーム



《写真２》文字が書かれた風船を鳥が割っていく



《写真３》キーを入力すると，鳥が風船へ向かっていく



《写真４》意外と難しく，高得点は出しにくい？

ゲームスピードの調整方法

プログラムリストの1200行に「<---Wait」と書いてある前の数値を増減させてください。０(ゼロ)にすると速く，数値を大きくすると遅くなります。

マシンによっては，この方法のウェイトではスピード調整がほとんど効かない場合があります。

最後に

何年ぶりだろうか。も

★付録CD復活求む！自宅のPCは未だスタンドアローンなので，是が非でも。（秋田県横手市・**ハングンマル20歳**）……**【編**：ベーマガのWebサイトの掲示板では「ベーマガにCD-ROMは必要か」なんていう論争が展開されていましたね。考えてはいるのですが…。**】**

のすごく久しぶりにPC9801でプログラムを組みました。今となってはWindowsにとって変わりましたが，プログラムは基本的に，この98全盛期と今とは何ら変わりないと思います。また，何か思いついたら投稿したいと思います。

CHECKER FLAG

影：ベーマガ投稿プログラムでは，まだまだ活躍しているPC98のDiskBASICの作品。基本はタイピングだけど，鳥が動いていく動作と風船の動きが，なかなかいいんじゃないかな？

Dr.D：DiskBASICとしては，若干長めのリストじゃが，コメントや字下げなど記述の仕方がていねいなリストじゃな。こういったリストの記述は，ぜひ他の投稿者にも見習ってほしいものじゃな。

つぐ美：掲載しているリストでは，文字コードの都合上，■の表記がDiskBASICとは異なっています。入力する時はちょっと注意してね。

《第1表》変数表

KARA%	鳥画像
KARAANI%	鳥アニメーション画像切り替え
HUX	風船の横軸
HUY	風船の縦軸
HUKEY	風船表示フラグ(値-1で非表示)
HUSEN%	英文字風船画像
GDAT%	画像作成用
DEM	風船縦揺れ用
KISO	風船の基礎位置
R	角度
HI	ハイスコア
WIT	ゲームスピード調整(ウェイト)
GRB＄	カラーパレット
LV	ゲーム難度
SCORE	ゲームスコア
ANI%	ゲームアニメーションフラグ
TIM	ゲーム残りタイム
HP	ミスカウント
KARAX	鳥横軸
KARAY	鳥縦軸
KARAR	鳥飛行角度
KARAV	飛行速度
HIT%	風船破裂判定用フラグ
I＄	キー入力用
ST	ミスタイプによる計算
KER%	ミスタイプ検出用
MOJI＄	風船文字
GX	文字無し風船の配置横座標
GY	文字無し風船の配置縦座標
DA＄	タイトル画像番号
HCK	風船の個数をチェック
A	鳥の移動方向計算用
FLG%	メッセージ表示タイミング用
I,X,Y	ループ
K	キー入力用
ST,SW	雑用

《リスト》「バルーンタイプ」プログラム・リスト

```
1000 '////////////////////////////////////////
1010 '          バルーンタイプ
1020 '
1030 '       FILE NAME : TYPEST.BAS
1040 '
1050 '     COPY RIGHT 2001 BY KEN-SOFT
1060 '
1070 '作成＆動作環境 : PC9801US(i80386SX-16MHz)
1080 '////////////////////////////////////////
1090 '
1100 '■■画面初期化■■
1110 *SHOKI
1120 CLEAR ,&H5E00 : PLAY ALLOC ,,,500,500,2000
1130 RANDOMIZE VAL(RIGHT$(TIME$,2))
1140 DIM KARA%(260,4) ,KARAANI%
1150 DIM HUX  ( 10)  ,HUY( 10) ,HUKEY( 10) ,HUSEN%(260,26)
1160 DIM GDAT%(260)  ,DEM( 26) ,KISO ( 26) ,R    ( 26)
1170 WIDTH 80,25 : CONSOLE 0,25 ,0 ,1 : COLOR ,,,,2
1180 SCREEN 3,0,0, 1 : CLS 3
1190 SCREEN 3,0,1,33 : CLS 3
1200 HI=100 : WIT=0 '<--Wait
1210 '
1220 '■■パレット設定■■
1230 *PAL
1240 GRB$="000002004006008000A00C00E72DA5F0900E0EE0666AAAEEE"
1250 FOR Y=0 TO 15
1260    COLOR=(Y ,VAL("&H"+MID$(GRB$ ,Y*3+1 ,3)))
1270 NEXT Y
1280 '
1290 '■■キャラ作成■■
1300 *CHRMAKE
1310 LOCATE 0,24 : COLOR 2
1320 PRINT "CAPSキーをONに、カナキーをOFFにしてください."
1330 LOCATE 0,25
1340 FOR I=0 TO 8 : PRINT : NEXT
1350 LOCATE 0,25 : COLOR 6
1360 PRINT "キャラクター作成中です。しばらくお待ち下さい。";
1370 RESTORE 3650 : FOR I=0 TO 260 : READ A$ : KARA%(I,0)=VAL("&H"+A$) : NEXT
1380 RESTORE 3890 : FOR I=0 TO 260 : READ A$ : KARA%(I,1)=VAL("&H"+A$) : NEXT
1390 PUT(100,  0),KARA%( 0,  0),PSET
1400 PUT(100, 32),KARA%( 0,  1),PSET
1410 FOR X=0 TO 31
1420    GET(131-X,  0)-(131-X, 63),GDAT% : PUT( X,  0),GDAT%,PSET
1430 NEXT
1440 GET( 0,  0)-( 31, 31),KARA%(0,2)
1450 GET( 0, 32)-( 31, 63),KARA%(0,3)
1460 RESTORE 3410 : FOR I=0 TO 260 : READ A$ : GDAT%(I)=VAL("&H"+A$)   : NEXT
1470 CLS 2
1480 MOJI$="ＡＢＣＤＥＦＧＨＩＪＫＬＭＮＯＰＱＲＳＴＵＶＷＸＹＺ"
1490 FOR I=0 TO 25 : PUT((I MOD 5)*32,INT(I/5)*32),GDAT%,PSET
1500    GX=(I MOD 5)*32 : GY=INT(I/5)*32
1510    PUT(GX+300,GY),KANJI(VAL("&H"+JIS$(KMID$(MOJI$,I+2,1)))),PSET
1520    FOR Y=0 TO 15 : FOR X=0 TO 15
1530          IF POINT(GX+300+X,GY+Y)<>0 THEN PSET(GX+X+9,GY+Y+6),0 : PSET(GX+X+8,GY+Y+5),15
1540 NEXT : NEXT : GET(GX,GY)-(GX+31,GY+31),HUSEN%(0,I) : NEXT
1550 IF INKEY$<>"" GOTO 1550
1560 GOTO *TITLE
1570 '
1580 '■■タイトル■■
1590 *TITLE
1600 SCREEN ,,0, 1 : CLS 3 : LINE(  0,  0)-(639,399), 8,BF
1610 SCREEN ,,1,33 : CLS 3 : LINE(  0,  0)-(639,399), 8,BF
1620 ANI%=0
1630 LOCATE 20,17 : COLOR 7 :PRINT "[1]  ゲーム開始  "
1640 LOCATE 44,17           :PRINT "[2]  ゲーム終了  "
1650 GET( 0,  0)-( 31, 31),GDAT%
1660 DA$="010011111414139919241504100413991814 0519"
1670 FOR I= 0 TO 11 : R(I)=RND*360 : DEM(I)=120 : KISO(I)=120 : NEXT
1680 FOR I=12 TO 19 : R(I)=RND*360 : DEM(I)=320 : KISO(I)=320 : NEXT
1690 COLOR 7 : LOCATE 30,25 : PRINT "PLEASE SELECT NUMBER";
1700 SCREEN ,,ANI% MOD 2
1710 LINE(128, 80)-(513,399), 8,BF
1720 FOR I=0 TO 19
```

PC-9801 N88-BASIC

入力方法は34ページをご覧ください。

★アホな受験生の法則：PS2を親に壊される。未だにPC-9801FXを使用し，学校にあった幻のN88-BASICのコンパイラ版に感動している。（長野県上田市・**LLL夫15歳**）……【**つぐ美**：壊されたり，感動したりのなか，受験のほうはどうだったのかな？】

```
1730    DEM(I)=10*SIN(R(I)*(3.14/180))+KISO(I)
1740    SW    =VAL(MID$(DA$,I*2+1,2))
1750    IF SW<>99    THEN IF I<12 THEN GOSUB *GP1 ELSE GOSUB *GP2
1760    IF R(I)>=330 THEN R(I)=0  ELSE R(I)=R(I)+30
1770 NEXT
1780 SCREEN ,,,(ANI% MOD 2)*32+1
1790 COLOR@(29,24)-(49,24),ANI%
1800 IF ANI% >= 7   THEN ANI%=0 ELSE ANI%=ANI%+1
1810 K=VAL(INKEY$)
1820 IF K>=1 AND K<=2 THEN ON K GOTO *START,*ENDGAME
1830 GOTO 1700
1840 '
1850 *GP1 : PUT(I*32+128      ,DEM(I)),HUSEN%(0,SW),PSET : RETURN
1860 *GP2 : PUT((I-12)*32+192,DEM(I)),HUSEN%(0,SW),PSET : RETURN
1870 '
1880 '■■スタート処理■■
1890 *START
1900 CLS
1910 FOR I=0 TO 1
1920    SCREEN ,,I
1930    LINE(  0,   0)-(639,399),  8,BF
1940    LINE(  0,   0)-(639, 16),  7,BF,15
1950    LINE(  0,383)-(639,399),  7,BF,15
1960    SCREEN ,,,I*32+1
1970 NEXT
1980 LOCATE  0,  0 : COLOR 0 : PRINT "TYPEING GAME"
1990 LOCATE 40,  0 :           PRINT USING "HI-SCORE ###0";HI
2000 LOCATE 30,  0 :           PRINT "LEVEL"
2010 LOCATE 60,  0 :           PRINT "SCORE"
2020 LOCATE 30,25 :           PRINT "TIME";
2030 LOCATE 60,25 :           PRINT "HP";: COLOR 5 : PRINT "■■■■■";
2040 COLOR 0
2050 FOR I=0 TO 9 : HUKEY(I)=-1 : NEXT
2060 ST=0        : LV=1       : SCORE=0    : ANI%=0  : TIM=500   : HP=10
2070 KARAX=320 : KARAY=200 : KARAR=0 : KARAH%=0 : KARAV=50 : HIT%=-1
2080 KARAANI%=0
2090 IF INKEY$<>"" GOTO 2090
2100 '
2110 '■■メイン処理■■
2120 *MAIN
2130 GOSUB *KEYHANTEI
2140 GOSUB *MOVE
2150 GOSUB *ATARI
2160 GOSUB *SCR
2170 GOSUB *LVUP
2180 GOSUB *ENDING
2190 FOR I=0 TO WIT:NEXT
2200 IF ANI%>=25 THEN ANI%=0 ELSE ANI%=ANI%+1
2210 GOTO *MAIN
2220 '
2230 '■■スコア計算＆表示■■
2240 *SCR : LOCATE 69, 0 : PRINT USING "###0";SCORE
2250        LOCATE 37, 0 : PRINT USING "##";LV
2260        LOCATE 37,25 : PRINT USING "###";TIM;
2270        COLOR@(62+HP,24)-(72,24),2
2280        TIM=TIM-1
2290 RETURN
2300 '
2310 '■■キー判定■■
2320 *KEYHANTEI
2330 I$=INKEY$
2340 IF I$<>"" THEN ST=ASC(I$)-65 ELSE ST=-1
2350 IF ST=-1  THEN RETURN
2360 KER%=0
2370 FOR I=0 TO  9
2380    IF HUKEY(I)=ST THEN KER%=-1 : HIT%=I
2390 NEXT
2400 IF KER%=0 THEN TIM=TIM-5 : HP=HP-1 : GOSUB *SE2
2410 RETURN
2420 '
2430 '■■鳥と風船の当たり判定■■
2440 *ATARI
2450 IF HIT%<=-1 THEN RETURN
2460 IF KARAX+15>HUX(HIT%)-10 AND KARAX+15<HUX(HIT%)+42 AND KARAY+15>HUY(HIT%)-10 AND
KARAY+15<HUY(HIT%)+42 THEN HUKEY(HIT%)=-1 : SCORE=SCORE+1 : HIT%=-1 : GOSUB *SE1
2470 RETURN
2480 '
2490 '■■風船＆鳥移動■■
2500 *MOVE
2510 SCREEN ,,ANI% MOD 2
2520 GOSUB *OTHR1
2530 HCK=0
2540 FOR I=0 TO  9
2550    IF HUKEY(I)<>-1 THEN HCK=HCK+1
2560 NEXT
2570 FOR I=0 TO  9
2580    IF HUKEY(I)=26 AND HUY(I)<200 THEN HUKEY(I)=INT(RND*26)
2590    IF HUKEY(I)>-1 AND HUY(I)< 50 THEN HUKEY(I)=-1 : SCORE=SCORE-1
2600    IF HUKEY(I)>-1 THEN HUY(I)=HUY(I)-(LV+2) : HUX(I)=HUX(I)+INT(1*COS(HUY(I)*3.14/180)) :
GOSUB *DISP1
2610    IF HUX(I)>607  THEN HUX(I)=HUX(I)-607
2620    IF HUX(I)<  0  THEN HUX(I)=607+HUX(I)
2630    IF INT(RND*50)<>0 AND HCK>0 GOTO 2650
2640    IF HUKEY(I)=-1 THEN HUKEY(I)=INT(RND*27) : HUX(I)=(RND*16+2)*32 : HUY(I)=351 : HCK=1 :
GOSUB *DISP1
2650 NEXT
2660 GOSUB *DISP2
2670 SCREEN ,,,(ANI% MOD 2)*32+1
2680 RETURN
2690 '
2700 '■■表示■■
2710 *DISP1
2720 IF HUKEY(I)= 26 THEN PUT(HUX(I),HUY(I)),HUSEN%(0,ANI%     ),PSET : RETURN
2730 IF HUKEY(I)<>-1 THEN PUT(HUX(I),HUY(I)),HUSEN%(0,HUKEY(I)),PSET
2740 RETURN
2750 '
2760 *DISP2
2770 IF HIT%<=-1 GOTO 2860
2780 IF HUX(HIT%)=KARAX THEN IF KARAY>HUY(HIT%) THEN A=90 ELSE A=-90 ELSE IF KARAX>HUX(HIT%)
THEN R=ATN((KARAY-HUY(HIT%))/(KARAX-HUX(HIT%))) : A=R*180/3.1416+180 ELSE R=ATN((HUY(HIT%)-
KARAY)/(HUX(HIT%)-KARAX)) : A=R*180/3.1416
2790 IF A<-180 THEN A=A+360
2800 IF A> 180 THEN A=A-360
2810 IF A>90 AND A<=270 THEN KARAH%=0 ELSE KARAH%=1
2820 KARAR=A/180*3.146
2830 KARAX=KARAV*COS(KARAR)+KARAX
2840 KARAY=KARAV*SIN(KARAR)+KARAY
2850 '
2860 PUT(KARAX,KARAY),KARA%(0,KARAANI%+KARAH%*2),PSET
2870 KARAANI%=KARAANI% XOR 1
```


★2台目のノートを買おうと考えているが，なかなか決まらない。今月の特集（CPU）は大いに参考になったが，まだペンティアムか，安価版にするか決められない。（福岡県行橋市・**ケンドー野心28歳**）……【**影**：私はコクヨがおすすめ。**つぐ美**：ノートちがいでしょ。】

```
2880 RETURN
2890 '
2900 '■■レベルアップ■■
2910 *LVUP
2920 IF SCORE MOD 20 = 0 AND LV*20<SCORE AND LV<16 THEN LV=LV+1 : LOCATE  0,25 : PRINT " LEVEL
UP !"; : FLG%=0 : SCORE=SCORE+10
2930 IF FLG%>=20 THEN FLG%=0 : LOCATE  0,25 : PRINT SPC(30); ELSE FLG%=FLG%+1
2940 RETURN
2950 '
2960 '■■効果音■■
2970 *SE1 :  PLAY ,,,"MBT150L32V1505CG>CR"                    : RETURN
2980 *SE2 :  PLAY ,,,,"MBT150L32V1504CCC"                     : RETURN
2990 *SE3 :  PLAY ,,,"MBT125L32V1405CDEFDEFGEFGAFGAB>C2R2"    : RETURN
3050 '
3060 '■■その他の処理■■
3070 *OTHR1  : LINE(  0,  17)-(639,382), 8,BF : RETURN
3080 '
3090 '■■ゲームオーバー■■
3100 *ENDING
3110 IF TIM>0 AND HP>0 THEN RETURN
3120 COLOR 7
3130 SCREEN ,,1,33 : CLS 3 : LINE(  0,   0)-(639,399), 8,BF
3140 SCREEN ,,0, 1 : CLS 3 : LINE(  0,   0)-(639,399), 8,BF
3150 IF HI < SCORE THEN HI=SCORE : LOCATE 32,11 : PRINT "NEW HI-SCORE !!"
3160 LOCATE 34,13 : PRINT USING "SCORE ###0";SCORE
3170 LOCATE 25,23 : PRINT       "TO BE CONTINU PUSH [SPACE] KEY"
3180 ANI%=0
3190 DA$="06001204991421041 7"
3200 GOSUB *SE3
3210 SCREEN ,,ANI% MOD 2
3220 LINE(176, 50)-(463,200), 8,BF
3230 FOR I=0 TO 8
3240    DEM(I)=10*SIN(R(I)*(3.14/180))+KISO(I)
3250    SW     =VAL(MID$(DA$,I*2+1,2))
3260    IF SW <>  99 THEN PUT(I*32+176,DEM(I)),HUSEN%(0,SW),PSET
3270    IF R(I)>=330 THEN R(I)=0 ELSE R(I)=R(I)+30
3280 NEXT
3290 SCREEN ,,,(ANI% MOD 2)*32+1
3300 COLOR@(24,22)-(54,22),ANI%
3310 IF ANI% >= 7     THEN ANI%=0 ELSE ANI%=ANI%+1
3320 IF INKEY$ <> " " GOTO 3210    ELSE *TITLE
3330 '
3340 '■■ゲーム終了■■
3350 *ENDGAME
3360 SCREEN 3,0,1,33 :  CLS 2
3370 SCREEN 3,0,0, 1 :  CLS 3
3380 END
3390 '
3400 '■■風船データ■■
3410 DATA 20  ,20  ,0   ,0   ,0   ,0   ,300 ,C0  ,FFFF,FFFF,300 ,C0
3420 DATA 0   ,0   ,1C00,38  ,FFFF,FFFF,1F00,F8  ,0   ,0   ,E000,7
3430 DATA FFFF,FFFF,FF00,FF  ,0   ,0   ,3   ,C000,FFFF,FFFF,FF03,C0FF
3440 DATA 0   ,0   ,4   ,2000,FFFF,FFFF,FF07,E0FF,0   ,0   ,8   ,1000
3450 DATA FFFF,FFFF,FF0F,F0FF,0   ,0   ,10  ,800 ,FFFF,FFFF,FF1F,F8FF
3460 DATA 0   ,0   ,20  ,600 ,FFFF,FFFF,FF3F,FCFF,0   ,0   ,40  ,200
3470 DATA FFFF,FFFF,FF3F,FCFF,0   ,0   ,40  ,200 ,FFFF,FFFF,FF7F,FEFF
3480 DATA 0   ,0   ,80  ,100 ,FFFF,FFFF,FF7F,FEFF,0   ,0   ,80  ,100
3490 DATA FFFF,FFFF,FF7F,FEFF,0   ,0   ,80  ,100 ,FFFF,FFFF,FF7F,FEFF
3500 DATA 0   ,0   ,80  ,100 ,FFFF,FFFF,FF3F,FCFF,0   ,0   ,40  ,200
3510 DATA FFFF,FFFF,FF3F,FCFF,0   ,0   ,40  ,200 ,FFFF,FFFF,FF3F,FCFF
3520 DATA 0   ,0   ,40  ,200 ,FFFF,FFFF,FF1F,F8FF,0   ,0   ,20  ,400
3530 DATA FFFF,FFFF,FF1F,F8FF,0   ,0   ,20  ,400 ,FFFF,FFFF,FF0F,F0FF
3540 DATA 0   ,0   ,10  ,800 ,FFFF,FFFF,FF07,E0FF,0   ,0   ,8   ,1000
3550 DATA FFFF,FFFF,FF03,C0FF,0   ,0   ,4   ,2000,FFFF,FFFF,FF00,FF
3560 DATA 0   ,0   ,3   ,C000,FFFF,FFFF,1F00,F8  ,0   ,0   ,E000,7
3570 DATA FFFF,FFFF,300 ,C0  ,0   ,0   ,1C00,38  ,FFFF,FFFF,0   ,0
3580 DATA 0   ,0   ,300 ,C0  ,FFFF,FFFF,0   ,0   ,0   ,0   ,0   ,80
3590 DATA FFFF,FFFF,0   ,0   ,0   ,0   ,0   ,40  ,FFFF,FFFF,0   ,0
3600 DATA 0   ,0   ,0   ,80  ,FFFF,FFFF,0   ,0   ,0   ,0   ,100 ,0
3610 DATA FFFF,FFFF,0   ,0   ,0   ,0   ,0   ,80  ,FFFF,FFFF,0   ,0
3620 DATA 0   ,0   ,100 ,0   ,FFFF,FFFF,FFFF,FFFF,0   ,100 ,0   ,0
3630 '
3640 '■■鳥１(左向き)■■
3650 DATA 20  ,20  ,0   ,300 ,0   ,0   ,0   ,300 ,FFFF,FFFF,60  ,D00
3660 DATA 0   ,200 ,60  ,F00 ,FFFF,FFFF,98  ,3100,60  ,E00 ,F8  ,3F00
3670 DATA FFFF,FFFF,86  ,4100,78  ,3E00,FE  ,7F00,FFFF,FFFF,81  ,4100
3680 DATA 7E  ,3E00,FF  ,7F00,FFFF,FFFF,8080,8100,7F  ,7E00,80FF,FF00
3690 DATA FFFF,FFFF,4040,8100,803F,7E00,C07F,FF00,FFFF,FFFF,A0A1,171
3700 DATA 405E,FE00,E0FF,FF71,FFFF,FFFF,7F92,289 ,806D,FC70,FFFF,FEF9
3710 DATA FFFF,FFFF,8A  ,70A ,FF75,F8F1,FFFF,FFFB,FFFF,FFFF,3186,98A
3720 DATA CE79,F671,CEFF,FF7B,CEFF,FF7F,7B81,11D4,847E,EE23,84FF,FF37
3730 DATA 84FF,FF3F,7B81,1D4 ,847E,FE23,84FF,FF37,84FF,FF3F,5A81,1D8
3740 DATA 847E,FE27,A5FF,FF3F,A5FF,FF3F,3141,190 ,CE3E,FE6F,CE7F,FF7F
3750 DATA CEFF,FF7F,A2  ,208 ,FF5D,FCF7,FFFF,FEFF,FFFF,FFFF,7F92,6C8
3760 DATA 806D,F837,FFFF,FEFF,FFFF,FFFF,2089,A90 ,FF76,F4EF,FFFF,FEFF
3770 DATA FFFF,FFFF,FF86,12E0,79  ,EC1F,FFFF,FEFF,FFFF,FFFF,F81 ,2290
3780 DATA FF7E,DCEF,FFFF,FEFF,FFFF,FFFF,742 ,4408,FF3D,B8F7,FF7F,FCFF
3790 DATA FFFF,FFFF,3F22,8489,FF1D,78F6,FF3F,FCFF,FFFF,FFFF,E52 ,48E
3800 DATA FF2D,F8F1,FF7F,FCFF,FFFF,FFFF,3D4E,868 ,FF31,F0F7,FF7F,F8FF
3810 DATA FFFF,FFFF,1F22,3088,FF1D,C0F7,FF3F,F0FF,FFFF,FFFF,312 ,C0E8
3820 DATA FF0D,F7  ,FF1F,C0FF,FFFF,FFFF,1F0A,C9  ,FF05,F6  ,FF0F,FF
3830 DATA FFFF,FFFF,306 ,8E  ,FF01,F0  ,FF07,FE  ,FFFF,FFFF,8701,30
3840 DATA 7F00,C0  ,FF01,F0  ,FFFF,FFFF,7F00,C0  ,300 ,0   ,7F00,C0
3850 DATA FFFF,FFFF,0   ,0   ,2000,20  ,FC01,F0  ,FFFF,FFFF,0   ,0
3860 DATA 0   ,0   ,4B02,28  ,FFFF,FFFF,FFFF,FFFF,28  ,4B02,0   ,0
3870 '
3880 '■■鳥２(左向き)■■
3890 DATA 20  ,20  ,0   ,0   ,0   ,0   ,0   ,0   ,FFFF,FFFF,0   ,0
3900 DATA 0   ,0   ,0   ,0   ,FFFF,FFFF,0   ,0   ,0   ,0   ,0   ,0
3910 DATA FFFF,FFFF,0   ,0   ,0   ,0   ,0   ,0   ,FFFF,FFFF,0   ,0
3920 DATA 0   ,0   ,0   ,0   ,FFFF,FFFF,20  ,6000,0   ,0   ,20  ,6000
3930 DATA FFFF,FFFF,50  ,9800,20  ,6000,70  ,F800,FFFF,FFFF,8089,471
3940 DATA 70  ,F800,80F9,FC71,FFFF,FFFF,7F96,428A,8069,BC71,FFFF,FEFB
3950 DATA FFFF,FFFF,AA  ,A10C,FF55,5EF3,FFFF,FFFF,FFFF,FFFF,3146,1188
3960 DATA CE39,EE77,CE7F,FF7F,CEFF,FF7F,7B81,C9D1,847E,362E,84FF,FF3F
3970 DATA 84FF,FF3F,7B81,31D6,847E,CE29,84FF,FF3F,84FF,FF3F,5A9D,DD8
3980 DATA 8462,F227,A5FF,FF3F,A5FF,FF3F,31A3,8290,CE5C,7C6F,CEFF,FE7F
3990 DATA CEFF,FF7F,42  ,4109,FF3D,BEF6,FF7F,FFFF,FFFF,FFFF,7F82,21C8
4000 DATA 807D,DE37,FFFF,FFFF,FFFF,FFFF,2081,1196,FF7E,EEE9,FFFF,FFFF
4010 DATA FFFF,FFFF,FF80,89E1,7F  ,761E,FFFF,FFFF,FFFF,FFFF,F81 ,6190
4020 DATA FF7E,9EEF,FFFF,FFFF,FFFF,FFFF,78E ,1908,FF71,E6F7,FFFF,FFFF
4030 DATA FFFF,FFFF,3F92,68C ,FF6D,F8F3,FFFF,FEFF,FFFF,FFFF,EA2 ,18B
4040 DATA FF5D,FEF4,FFFF,FFFF,FFFF,FFFF,3D42,C168,FF3D,3EF7,FF7F,FFFF
4050 DATA FFFF,FFFF,1F82,3188,FF7D,CEF7,FFFF,FFFF,FFFF,FFFF,382 ,DE8
4060 DATA FF7D,F2F7,FFFF,FFFF,FFFF,FFFF,1F86,42C8,FF79,BCF7,FFFF,FEFF
4070 DATA FFFF,FFFF,38A ,218E,FF71,DEF1,FFFB,FFFF,FFFF,FFFF,8791,9931
4080 DATA 7F60,66C0,FFF1,FFF1,FFFF,FFFF,7FA0,66C0,340 ,1800,7FE0,7EC0
4090 DATA FFFF,FFFF,40  ,1800,2000,20  ,FC41,18F0,FFFF,FFFF,0   ,0
4100 DATA 0   ,0   ,4B02,28  ,FFFF,FFFF,FFFF,FFFF,28  ,4B02,0   ,0
```

★ノートパソコン買って日々フリーズと戦闘中です。なんだMeもたいしたことないねえ…。（岩手県釜石市・**鉄人坂井23歳**）……
【**編**：Meを使っているなら，すぐXPにアップグレードしたほうがいいですね。影さんもですよ。**影**：じゃ編くん，XP買ってよ！】

FX-890P用



いわば熱帯魚すくい

熱帯魚（エンジェルフィッシュ）

八木　克彦（34歳）

内容

熱帯魚が大好きな魚吉くん。遠いジャングルの果てまで，網を片手に熱帯魚採集に日本を旅立った。ジャングルに着いた魚吉くん，そこには探し求めていた熱帯魚が群れをなして泳いでいた。さぁ魚吉くん，体力の続く限り熱帯魚採集だ。

遊びかた

プレイヤーは魚吉くん（画面下に表示されるキャラクター）を左右に動かし，ちょうど真上に熱帯魚（〈,〉で示される）が来たときにキーを押すと網がでて，熱帯魚を捕まえることができます。一度にたくさんの熱帯魚を捕まえるとボーナス得点が入ります。

操作方法

プレイヤーは，数字キーの“1”で左，“2”で右，“Z”キーで網をだします。プレイヤーの前にはライン“-”が引いてあり，一度網をだしたところには“*”が表示され，次はそこから網はだせません。“*”で手前ラインがいっぱいになり網をだすことができなくなるとゲームオーバーになります。また，画面の右端下側には体力(POWER)の残りが表示されており，体力がなくなってもゲームオーバーです。体力はゲームクリアごとに100ずつ回復します。熱帯魚の数が少なくなってくると，熱帯魚が激しく泳ぎだしプレイヤーの体力もはやく減りだします。熱帯魚は一匹ずつ捕まえるより，まとめて捕まえれば高得点となります。

改造のアドバイス

IF文が多く,処理が少し遅いようなので，IF文を上手に減らしてみてください。プログラムリストの10行のFの値を変えると熱帯魚の数を増減できませす。

380行のC=509-Cはプレイヤーキャラクターが左右に向きを交互に変える処理をしています。DEFCHR$()でキャラクター定義されたコード254と255を足すと509となります。最初にCを254とすると，C=509-Cは509-254でCは255となります。次にC=509-Cを実行すれば，509-255でCは254に戻ります。この引き算の処理によってプレイヤーキャラクターが向きを変化させています。単純なアイデアですが，けっこう気にいっています。

苦労した点

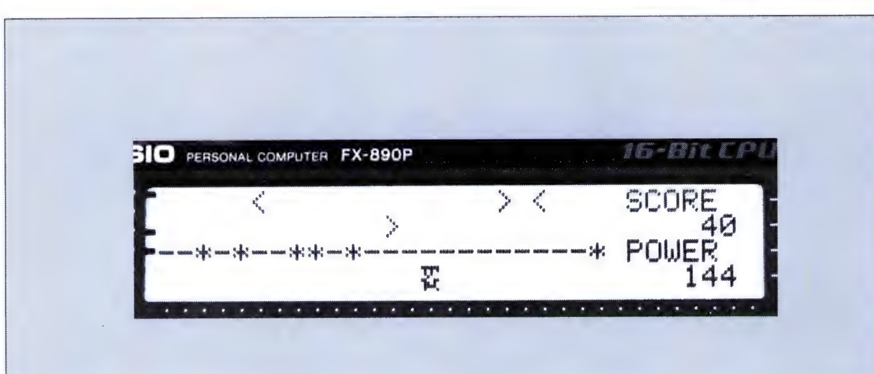
この作品の制作そのものには，とくに苦労はしませんでしたが，頭の中で描いていたいろいろなアイデアがまとまるまで時間がかりました。

コメント

前作『Quick Timer』が没になったので，名誉挽回のため送り出した最新作です。よろしくお願いします。熱帯魚は乱数で動いていますが，ときにはまるで考えているかのような動作をします。結構手強く，体力がなくなってゲームオーバーとなることもしばしばです。

CHECKER FLAG

つぐ美：ポケコンらしいゲーム作品なんじゃないかしら？ポケコンの横長の画面をうまく活用している印象です。
編：乱数メインの熱帯魚の動きですが，意外とそれらしく見えるから不思議です。
Dr.D：このゲームでは，プレイヤーの目の前の熱帯魚しかとれないが，網の処理を行うルーチンをもう一つ作り表示座標を変更すれば，奥の熱帯魚が捕獲できる網というのが作成できるゾ。興味をもったポケコンユーザーはぜひとも挑戦してみてほしい。



《写真１》泳ぎ回る熱帯魚を捕まえていく

★2月号84ページ，HDDを10GBから80MBへとは，すさまじいダウングレードでんな。これじゃ，Windowsすら入らないんじゃ？（って，素直に誤植発見ってかけよ）。（大阪府門真市・上7 28歳）……【編：はっ！ごめんなさい。80GBの誤りでした。申し訳ありません。】

《第1表》変数表

X	プレイヤーのX座標
XW	プレイヤーの一つ前のX座標
X()	熱帯魚の現在のX座標配列
Y()	熱帯魚の現在のY座標配列
XW()	熱帯魚の一つ前のX座標配列
YW()	熱帯魚の一つ前のY座標配列
F	熱帯魚の最大の数
F()	熱帯魚が動作中かどうかの配列
L	熱帯魚の残りの数
G	一度につかまえた熱帯魚の数
C	プレイヤーキャラクタ
C()	熱帯魚のキャラクター配列
D()	熱帯魚の移動方向配列
M	網の出せる位置の残り，(*)ではない所
S	得点スコア
P	パワーの残り
K$	キー入力
T$	土手のライン
A	ループ用



《第2表》行ごとの処理説明

10-20	初期化１，配列の定義，プレイヤーキャラクター定義
30	初期化２
40-50	ゲームスタート待ちループ
60-110	熱帯魚の初期化
…60	表示画面等の初期化
70-110	熱帯魚の位置，キャラクター，向きなどの設定
120-160	スコア，パワーの表示
170-350	メインルーチン
180	その熱帯魚がすでにつかまっているか
190-200	乱数により熱帯魚の移動方向を変化させる
210-240	熱帯魚の移動，キャラクターの向きを変更する
250-270	熱帯魚の消去と表示
280	キー入力
290-310	キー入力によって無視または左右へ移動
320	キー入力が“Z”または“z”なら*SUBへ
330-340	プレイヤーキャラクターの消去と表示
360	パワーを減らす，パワーの残りの表示
370	パワーが無くなったらゲームオーバー処理へとぶ
380	プレイヤーキャラクターの向きを変えてメインルーチンへ
390-500	*SUB　サブルーチン
…390	指定された位置に“*”があれば戻る
…400	指定された位置から網をだす
…410-430	網につかまった熱帯魚を数える
…440-450	指定された位置に“*”を表示する
…460	得点計算など
…470-480	得点の表示
…490	ゲームオーバー処理
…500	ゲームクリア処理，パワーの回復

FX-890P

入力方法は34ページをご覧ください。

《リスト》「熱帯魚」のプログラム・リスト

```
10 CLS:CLEAR:F=7:DIM X(F),Y(F),XW(F),YW(F),C(F),D(F),F(F)
20 DEFCHR$(254)="88EF96E28EC9":DEFCHR$(255)="C98EE292EF88"
30 X=23:XW=X:C=254:S=0:G=0:P=200
40 LOCATE 8,2:PRINT " PUSH SPACE KEY!";
50 IF INKEY$<>"       " GOTO 50
60 M=24:L=F+1:CLS:T$="------------------------"
70 FOR A=0 TO F
80 X(A)=INT(RAN#*8):Y(A)=INT(RAN#*2):C(A)=62:D(A)=1:F(A)=1
90 LOCATE X(A),Y(A):PRINT CHR$(C(A));
100 XW(A)=X(A):YW(A)=Y(A)
110 NEXT
120 LOCATE 25,0:PRINT "SCORE";
130 LOCATE 25,1:PRINT USING "######";S;
140 LOCATE 25,2:PRINT "POWER";
150 LOCATE 25,3:PRINT USING "######";P;
160 LOCATE 0,2:PRINT T$;
170 FOR    A=0 TO F
180 IF F(A)=0 GOTO 350
190 IF INT(RAN#*30)>=28          THEN D(A)=-D(A)
200 IF INT(RAN#*30)>=28          THEN Y(A)=1-Y(A):D(A)=-D(A):GOTO 240
210 X(A)=X(A)+D(A)
220 IF X(A)<0 THEN X(A)=0:D(A)=-D(A)
230 IF X(A)>23 THEN X(A)=23:D(A)=-D(A)
240 IF D(A)<0 THEN C(A)=60 ELSE  C(A)=62
250 LOCATE XW(A),YW(A):PRINT " ";
260 LOCATE X(A),Y(A):PRINT CHR$(C(A));
270 XW(A)=X(A):YW(A)=Y(A)
280 K$=INKEY$
290 IF K$="" GOTO 330
300 IF K$="1" AND X>0 THEN X=X-1
310 IF K$="2" AND X<23 THEN X=X+1
320 IF K$="Z" OR K$="z"          THEN GOSUB *SUB
330 LOCATE XW,3:PRINT "                ";:XW=X
340 LOCATE X,3:PRINT CHR$(C);
350 NEXT
360 P=P-1:LOCATE 25,3:PRINT USING "######";P;
370 IF P<=0 THEN M=0:GOSUB 460
380 C=509-C:GOTO 170
390 *SUB:IF MID$(T$,X+1,1)="*" THEN RETURN
400 LOCATE X,1:PRINT CHR$(79);CHR$(29);CHR$(31);CHR$(124);
410 FOR    A=0 TO F
420 IF X(A)=X AND Y(A)=1 AND F(A)=1 THEN G=G+1:F(A)=0:L=L-1:BEEP:LOCATE        X(A),1:PRINT " ";
430 NEXT
440 T$=LEFT$(T$,X)+"*"+RIGHT$(T$,23-X)
450 LOCATE 0,2:PRINT T$;
460 S=S+G^3*10:G=0:M=M-1
470 LOCATE 25,1:PRINT USING "######";S;
480 LOCATE X,1:PRINT " ";
490 IF M<=0 THEN LOCATE          10,2:PRINT " GAME OVER!";:WAIT 50:RETURN 30
500 IF L=0 THEN        P=P+100:BEEP:BEEP:RETURN 60 ELSE RETURN
```

★最近，ブロードバンドってやたら言うけど，実感がないです。（鹿児島県川内市・**ちゃっぴい28歳**）……【**編**：たしかに，体験してみないことには，わかりませんよね。**影**：ただ待っていても仕方ないので，自分からどんどん新しいものを取り入れていくことが大切。】

★★★★★★★ マイコンBASICマガジン ★★★★★★★

明日のスター・プログラマー

Programmer's Room

今月のベスト・プログラマー

編：投稿数はもう一歩といった感じの今月でしたが，採用作品はなかなか完成度の高いプログラムが揃ったという印象です。特に今月は，連載講座も好調なHSPの作品が目に付きました。

影：今月は作品がなかったけど，VBA作品など連載講座のある言語は，その連載を執筆されているライターさんがけっこう注目していたりするゾ。投稿したプログラムをライターさんが気に入ってくれれば，連載講座の中で取り上げてくれることがあるかもしれないな。

つぐ美：そういう展開が広がってくると，投稿プログラムコーナーももっと盛り上がってくるかもしれませんね。

さて，お待たせしました，本題のベストの選考といきましょう。

影：移植版ではあるけど，iアプリの『iエジプトの石』が印象に残っている。VB版の画面をしっかり再現しているのは，インパクトがあったなぁ。

編：アイデア面で見ると，やはりHSPの２作品が良いですね。特に『Rolling Panels』は，ゲームアイデアだけでなく，リスト面でもしっかり作られているので，まさにベーマガらしい作品と言えるのではないでしょうか？

つぐ美：アイデア面では，他にもVBの『あーる・かるきゅれーしょん Ver1.1』など個性的な作品がありますが『Rolling Panels』は，そのアイデア＋リスト面で抜き出ていますからね。これはもうベストに選んでもいいのではないでしょうかね，Dr.？

Dr.D：うむ，ワシとしても今月の採用作品の中では，『Rolling Panels』がベストにふさわしいと思うゾ。リスト面で久しぶりに感心した作品じゃったしな。

つぐ美：というわけで，今月のベストプログラマーは『Rolling Panels』の作者，斎藤和哉さんに決定です！！

影：さて，ベストの選考も無事に終わったところでDr.，今月は「採用へのウラ技」を紹介してもらいたいのですが。

Dr.D：うむ。特にこれは，まだ掲載されたことのない投稿者に目を通してほしい内容じゃな。中には本当にウラ技的な事項もあるが，これらを参考にぜひとも誌面へ掲載される喜びを味わってほしいものじゃ。

今月はもう一歩でした

●Visual Basic用『Hit&Blow』／林　泰広(31歳)

●Visual Basic用『Numbers!』／高　悠史(17歳)

●Java用『HIT&BLOW』／富田紘平(15歳)

Dr.D：この三つの作品すべてが，いわゆる数字当てゲーム。それぞれの作品ごとに独自の工夫があったりはするのじゃが，この数字当てゲーム自体が比較的簡単な構造で作れてしまうため，やはり掲載という面では言語やリスト面で抜き出た部分がないとキビシイということを言っておこう。これは今回の数字当てゲームに限らず，投稿の多いジャンルである暗号化ソフトやキャラクタエディタ，ブロック崩しやライツアウトなどの作品にも言えることじゃな。それでも上記のジャンルで投稿したいという読者は，その作品の完成度はもちろん，前述しているようにリスト面などでの工夫もしっかり行ってほしい。定番のジャンルなだけに中途半端な内容では掲載は難しいゾ。

●HSP用『Gun Sniper』／INF (15歳)

つぐ美：壁のはね返りを利用したシューティングゲーム。しっかり作られているのですが，超大作としてはチョット物足りないという印象でした。せっかく超大作で制作するのだから，演出面などにもこだわってみてはどうでしょうか？

投稿プログラム

「採用へのウラ技」

・説明ファイルをしっかり記述。
特に遊び方の項目をしっかりと。長すぎたり，極端に短い説明ファイルにも問題あり。掲載されている作品の説明文を参考に。

・投稿作品は自分オリジナルのもので。
特に過去に掲載された作品と，ほとんど同じ内容の作品の印象は最悪。これだけは避けたいところじゃ。また，権利での不採用を避けるため，画像やゲームデザインもなるべくオリジナルものにしよう。

・掲載数が少な目の言語を狙え。
WonderWitchやDirectX，C言語系は狙い目かもしれんゾ。

・入力しやすい作品(リストが短いなどの特徴がある作品)も狙い目。

・3Dモノのプログラムは，見た目のインパクトがバッチリ！！

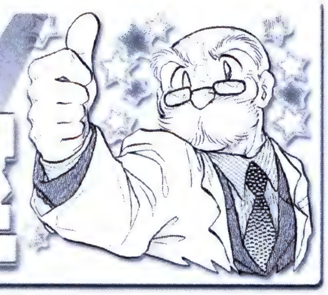
・毎月コンスタントに投稿する。
月によって投稿数は違ってくるので，運が良ければ，投稿数が少な目の月に巡り合う可能性もある？

・小中学生，女性からの投稿は，掲載されやすいかもしれんなぁ…？

投稿ありがとう

INF／富田紘平／高　悠史／サムセイド／阿部雅之／林　泰広／石山明弘／馬場章吾／奥田茂延／梶原寿樹／八木克彦／山澤正裕／斎藤和哉／ロディマス／三好利治／ていこく／Mr.親父／清家　巧／サル吉／ハングルマン／LLL夫／まぢけん／ケンドー野心／鉄人坂井／瑠璃／Gleam／四谷一獲
※全員紹介できなくてごめんなさい。

Dr.Dのベスト・プログラム研究室



今月のベスト・プログラム：HSP用『Rolling Panels』(62ページ掲載)

今月のベスト・プログラムは，HSP用『Rolling Panels』。作品のアイデアとリスト共に評価の高かった作品じゃ。特にこのコーナーでは，リスト面に重点をおいて解説をしてみよう。解説部分としては，やはりこの作品の動きの肝となっている回転部分に関する処理と，ゲーム進行に関わってくるマウス処理，そしてパネルの作成方法と，その処理を利用したパネルの変更を行う改造を紹介していこう。

(Dr.D)

パネルの回転処理

プログラムで回転の動きを処理するのに必ずと言っていいほど登場してくるのが三角関数じゃ。sin,cos命令を利用してx,y座標を求めていくことで円の動きを再現できることから，回転などの動きを処理するときに使用するのじゃ。本来ならばこの動きの原理さえ理解してしまえば，あとはプログラムでsin,cosを呼び出して利用するだけなのじゃが，実はHSPではこのsin,cos命令が用意されていない。

そこでこのsin,cos命令を自分で作成する必要がある。といっても汎用性のあるような関数の形で作成するのも大変なので，ここではsin,cosで求める値をもった配列を作成するようにしている。その処理を行っているのが**リスト1**の部分じゃ。sinカーブの0～90度までの値は自分で定義しておいて，あとはその値を利用して残りのsinの値とcosの値を作成しているわけじゃ。

この配列が作成できれば，あとはこれを利用して回転を実現する座標計算処理を記述すればOKじゃな。この座標計算は**リスト2**の部分で使用されておる。記述がHSP独特なのでチトわかりにくいかもしれんが，リストに追加記述したコメントを参考に実現方法をしっかり理解してほしい。

《リスト1》sin,cosの値を持つ配列を作成するリスト

```
; ---- [ 第1象限分の sin テーブル ]
st=0,89,178,265,350,433,512,587,658,724,784,839,887,928,962,989,1008,1020

; ---- [ sin, cos テーブルを作成 ]
repeat 18 : i = 35 - cnt : st.i = st.cnt : loop;            第2象限
repeat 36 : i = 36 + cnt : st.i = -st.cnt : loop;           第3,4象限
repeat 72 : i = 18 + cnt : if i>71 : i -= 72
 ct.cnt = st.i : loop;          cos テーブルの作成
```

《リスト2》回転処理を行うパネル表示部分のリスト

```
*_output
gsel 0 : redraw 2 : pos 0,0 : gcopy 2,0,0,440,440

repeat 36
;repeat 1にすると，表示パネルが1枚になり，回転の動きがより良くわかる

        ;aの値が角度の代わりとなる
        a = cnt * 2 + b : if a > 71 : a -= 72
        a = 71 - a

        ;このpos命令で回転を処理する座標計算を行う
        ;stがsin，ctがcosとなる。
        pos (m*st.a)/1024+206,206-((m*ct.a)/1024) : a = pl.cnt

        ;パネル表示
        gcopy 3,x.a,0
loop
if ans>=0 : pos 206,206 : gcopy 3,x.ans,0
redraw 1 : return
```

マウス関連処理

ゲームを行う上で必要となるのが，このマウス処理。とくにこのゲームはマウスカーソルで指定したパネルの判定を行うため，この部分の処理をしっかり記述する必要がある。このマウス処理は，**リスト3**の部分で行っておる。HSPではマウスの座標は簡単に得ることが出来るので，ここは問題ないじゃろう。

肝となるのは，正解パネルとの判定処理じゃな。難しそうに感じるかもしれんが，実はこのプログラムでは，正解パネルの現在の座標を求めて，その座標とマウスで指定した座標との位置を比較することで，正解パネルをクリックしたかどうかを判定しているのじゃ。

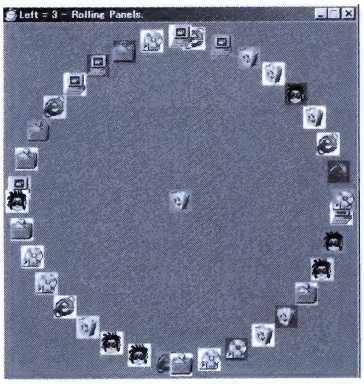
「何だ，基本的で単純な処理じゃん」と思うかもしれんが，これはリストの構造がしっかり出来ているからこそ実現できた方法であることを認識してもらいたい。下手なプログラムでは，正解パネルの現在の座標を得るという処理が非常に面倒になってしまうことがあるからじゃ。

パネルの作成方法

まず，パネルの作成方法を解説する前に，どのようなパネルを作成してい



▲《第1図》作成パネルの内容



◀HSP用「Rolling Panels」

るのかを見てもらいたい。**第1図**がこのプログラムで作成されるパネルの内容じゃ。１枚１枚のパネルが横長に配置されている画像を作成するわけじゃ。

DirectDrawなどのプログラムを経験している人ならピンとくるじゃろうが，このような画像を作っておけば，あとはこの画像から指定した部分のみを画面に転送していくという処理でパネル表示ができるわけじゃ。

このパネル作成を行うのが**リスト４**。あまりコメントは追加していないが，わかりやすいよう命令ごとに区切って記述してみたので，見やすいリストになっているハズじゃ。

改造してみよう

この前の項目で解説したパネル作成の処理を見て「作成するパネル画像の代わりにビットマップファイルなどが使えるんじゃないか？」と思った読者もいるのではないかな？その疑問に対する答えはYes。ビットマップファイルの使用は簡単な改造で可能になるのじゃ。

要はプログラム側で描画する図形の代わりに，パネルの大きさに合わせた画像ファイルを描画すればOK。ただ，注意が必要なのは画面モード。場合によってはこれを設定し直さないと画像が正しく表示されなくなる。

詳細はHSPのscreenやgmodeのヘルプを参照してほしい。改造を行った**リスト５**ならびに実行画面を掲載しているので，これを参考にぜひとも改造に挑戦してほしい。

今回は，普段よりもリストを掲載しての解説が多くなってしまったが，これもこの作品のリストが素晴らしい証拠でもある。普段HSPを使用していない投稿者も，この作品のリストには目を通しておいてほしいものじゃな。

《リスト３》マウス処理を行うリスト

```
*_click;

;マウスで指定した座標を得る
mx = mousex : my = mousey

;36枚の中から，正解パネルの番号を得る
repeat 36 : if pl.cnt==ans : i = cnt : break
loop

;正解パネルの現在の座標を求める
;現在の回転した角度(b)と正解パネルが何枚目(i)にあるか,
;という情報を用いて，座標を求めている
a = i * 2 + b : if a > 71 : a -= 72
a = 71 - a : px = (m*st.a)/1024+206 : py = 206-((m*ct.a)/1024)

;文字表示に関する設定
color 0,255,0 : font fname,64 :

;+表示処理。+は２本の線で表示している
;位置を決める
pos px - 18, py - 18
;縦棒表示
boxf mx-1,my-8,mx+1,my+8 :
;横棒表示
boxf mx-8,my-1,mx+8,my+1

;パネルの正解判定処理
;正解パネルのある座標とマウスで指定した座標を比較する
i = 2 : a = 28 + i; 許容誤差
if (mx<(px-i))|(my<(py-i)) : goto *_miss
if ((mx-px)>a)|((my-py)>a) : goto *_miss

;以下正解時の○表示処理(１部のみ掲載)
mes "○" : pos 206,206 : gcopy 2,0,0
```

《リスト４》パネル作成を行うリスト

```
;ブロック作成
buffer 3,1008,28,1 : pch = "○△□●▲■"

;ブロックの背景の色
;１パネルごとでなく，複数枚分まとめて背景色を表示させている
color 0,0,255 : boxf 0,0,335,27 :  ;背景が青
color 255,0,0 : boxf 336,0,671,27  ;背景が赤
color 255,255,255 : boxf 672,0,1007,27 : ;背景が白

font fname,27,1

;色を指定して，パネルの模様を表示
pos 0,0 : mes pch : pos 336,0 : mes pch
color 0,0,255 : pos 504,0 : mes pch + pch
color 255,0,0 : pos 168,0 : mes pch : pos 840,0 : mes pch
```

《リスト５》パネル表示の改造リスト

```
;ゲーム画面のバック
buffer 2,440,440,1 : color 128,128,128 : boxf 0,0,439,439

;改造部分
;panel.bmpは自分で作成した画像。1008×28の大きさ
buffer 3,1008,28,0
picload "panel.bmp",1

;パネルの１部分を変更したい時は,
;buffer 4,28,28,1
;picload "panel.bmp",1
;pos 0,0 : gcopy 4,0,0,28,28
;といった命令を，本来のパネル描画部分と組み合わせる

repeat 36 : x.cnt = cnt * 28 : loop

; ---- [ Title ]
*_title
;画像が正しく表示されるよう，画面モードの設定を変更する
;screen, gmodeのパラメータの1を0に変更する
;変更すると動作速度に影響が出る(動作が遅くなる)
 screen 0,440,440,0,(dispx - 446) / 2,(dispy - 460) / 2 : title ttl
 gmode 0,28,28 : font fname2,24,3 : objsize ,,32 : color 0,0,0
```



これから始める**ブロードバンド活用術**

ADSL HowTo!! 体験記

規格の話とPC環境&費用で選ぶ導入術

ISDN回線と比べてもADSLの導入はそれほど難しくなく，ケーブル接続と簡単なOSセットアップで，手軽に利用できるのが特徴です。これだけで数Mbpsもの高速なインターネット接続環境が手に入るのですから，ADSLを導入しない手はないですね!!　今回は，ADSLの専門用語の解説とパソコン環境と予算に合わせた機材選びを紹介します。通信料金の請求書が初めて送られてきたので，その話も少々…。

先月号のおさらいとまとめをしておきます。

ADSLの特徴として，

①ISDNやモデム接続とは比較にならない「高速通信」が可能
②一定料金で24時間使い放題の「常時接続」ができる
③上りと下りの回線速度が異なる「非対称通信」
④一般の電話回線（メタル・ケーブル）が利用できるため，導入費用は格安で「コストパフォーマンスは抜群」
⑤パソコンへは，「LAN接続」（イーサネット接続）が基本

などがあることだけは，覚えておきましょう。最近では，ADSLモデムの一部にパソコンの周辺機器増設端子でお馴染みのUSBを装備したモデルをあるようで，⑤の項目に「USB接続」を加えても良いでしょう。

さて，先月号の基礎講座の引き続きとして，ADSLにまつわる用語の解説をしておきましょう。購入または契約したプロバイダーからレンタルしたスプリッターやADSLモデムが揃えば，ケーブルの接続だけで，知識がなくてもADSLを利用することは可能ですが，本誌の読者であれば，ADSLに関しての知識やテクニックは知っておいてもらいたいものです。導入後のADSL活用にも必ず役に立つことなので，知っておいて損はしないはずです。

ADSL以外にもある電話線高速通信の方式

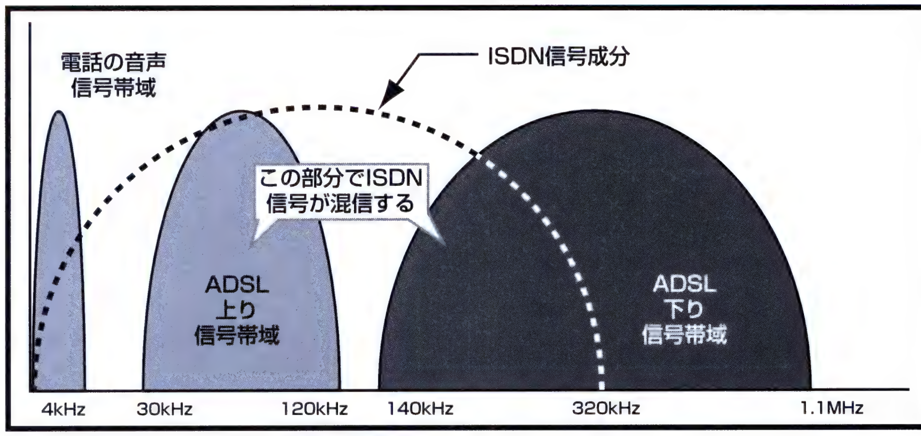
ADSL関連の資料には，「G.dmt」や

《第1表》xDSLのおもな種類と特徴

名称	ADSL (Asymmetric Digital Subscriber Line)	HDSL (High bit rate Digital Subscriber Line)	IDSL (ISDN Digital Subscriber Line)	SDSL (Symmetric Digital Subscriber Line)	VDSL (Very high bit rate Digital Subscriber Line)
接続方法	メタル電話回線(2線式)×1	メタル回線(4線式)×1	メタル電話回線(2線式)×1	メタル電話回線(2線式)×1	メタル電話線(2線式)×1
電話局との回線距離	4km程度まで	5km程度まで	不明	3km程度まで	1km程度まで
回線速度	上り512k～1.5Mbps 下り1.5M～8Mbps	下り・上りとも 1.5M～2Mbps	上り・下りとも 160kbps程度	上り・下りとも 1.5M～2Mbps程度まで	上り1M～2Mbps 下り10M～50Mbps
用途・特徴	機材の価格が安く一般アナログ電話が利用できるため導入費用が他方式に比べ安価・電話との共用OK	光ケーブルを使わないデジタル回線のビジネス向け。HDSLを2回線ペアで使用するSHDSL（Single pair HDSL）もある・データ通信専用で電話との共用は不可	ISDN技術を応用したデジタル通信専用回線で現在ほとんど利用されていない	ビジネス向け回線だがADSLと比べそれほどのメリットはない	光ケーブル回線のオフィスビルやマンションなどへの屋内引き込み配線向け。集合住宅などで今後普及が期待されている技術だ

「ANNEX C」など，これまで目にしたことのない用語がやたらに出てきて，チンプン・カンプン!という人も多いはず。まず，用語の整理からはじめましょう。

ADSLは，ツイスト・ペアすなわち2芯の電話線で高速通信を行う技術であるDSL(デジタル・サブスクライバー・ライン)の1つの種類で，電話局から加入者への下り回線とその逆の上り回線の通信速度が異なる非対称通信方式となっています。



《第1図》ISDNのADSL回線への影響

ADSL以外にも，DSLには**第1表**のように，「HDSL」や「SDSL」,「VDSL」などがあり使用環境などや用途ごとに使い分けがされています。頭のひと文字の違いから，DSLの総称を「xDSL」とする場合が多いようです。

なかでも，マンションなどの集合住宅やビジネス・ビルなどでのブロードバンド回線の導入に用いられ最近脚光を浴びている「VDSL」です。

建物の近くまで光ファイバーを引き，各家庭や事務所へは，建物内に張りめぐらされた既存の電話線をそのまま利用して高速通信が行えるのが，VSDLなのです。自宅がマンションの人や学校などで多数のインターネット回線が必要な場所では，知らないうちにVDSLのお世話になっている人もいるかもしれませんね。ただし，加入者側からは名称は異なるものの機材や接続方法はVDSLとADSLは同じといって良いでしょう。

通信速度と接続方式が異なる G.lite,G.dmt,ANNEX 規格

少々やっかいなのが，ADSL回線の規格と接続方式にまつわる用語です。「G.lite」や「G.dmt」,「ANNEX A」,「ANNEX C」などがそれです。

G.liteとG.dmtは，世界的に決められたADSL標準規格のことです。G.liteは「ハーフレイトADSL」や「簡易ADSL」とも呼ばれ，名称どおり比較的低速な通信速度に設定されています。国内で最初にサービスが始まった1.5Mbpsタイプの ADSLがこれにあたります。これよりも通信速度を速くした規格がG.dmtで，最近の8 Mbps接続サービスがそうです。通信速度の速さからG.dmtは「フルレイ

カード追加で無線ADSLルータにもなる低価格モデル お買い得なフルレートADSL対応の高速ルータ

ここのところ，ブロードバンド・ルータが大人気だ。1万円前後からそれ以下の低価格モデルが登場したことが要因だが，PC周辺機器でお馴染みのメルコから発売の，この「BLR2-TX4」(13,800円)もその1台。

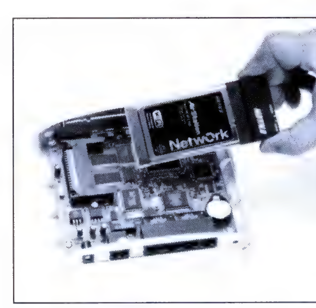
ルータ内部の処理速度が9.2Mbpsと高速で，8 MbpタイプADSLにも満足できる性能。さらに，内部にカードスロットがあり同社の無線LANカードの増設で，無線LANルータにもなる拡張性の良さも魅力のひとつ。小規模のビジネス向けLANにも耐えるコストパフォーマンスの優れたモデルといえるだろう。



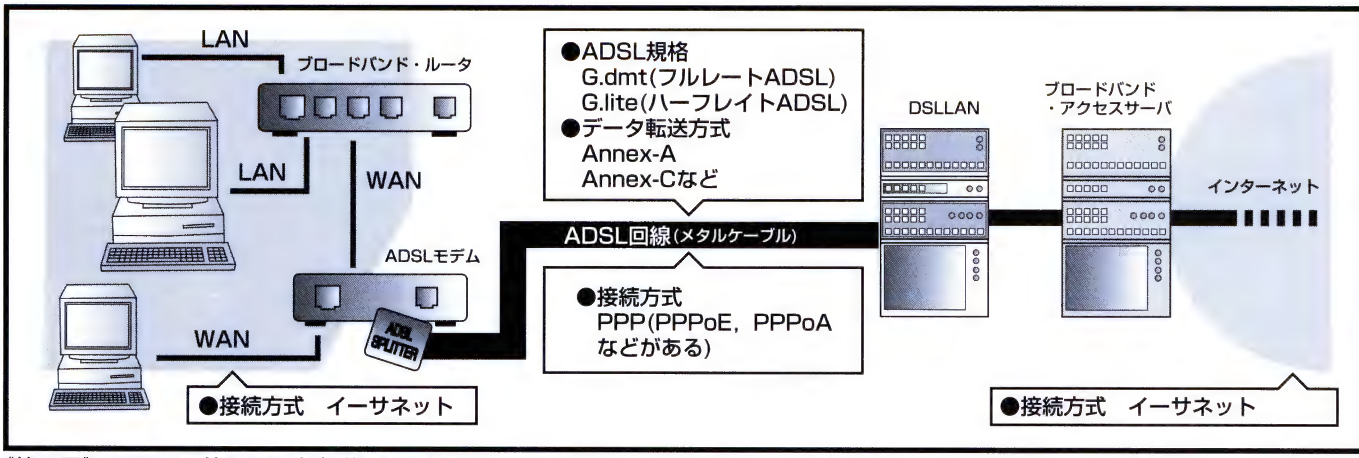
▲1万円そこそこでブロードバンドLANネットワークが構築できる魅力的な製品だ







▲本体内部のカードスロットへ同社の無線LANカード「WLI-PCM-L11G」をセットすることで無線LANルータに変身。無線LANカードをもう1枚用意すれば，ノートPCからもADSL回線にワイヤレスで接続できる。



《第２図》ADSLで使われる各部分の用語

ト ADSL」とも言われています。

「ANNEX A」,「ANNEX C」は,「G.lite」と「G.dmt」規格のなかに含まれる，ADSLの通信方式の名称です。ANNEX（アネックス）には，A，C，Hの３種類ありますが，おもに使われているのはAとCで，Aは最初にADSLが普及した米国内（ヨーロッパ各国も含む）の規格です。

ANNEX Cは，ANNEX Aをもとに日本国内向けに改良された規格となっています。実はこの違いには，米国と日本国内とのISDN回線の接続方式の違いが大きく関わっているのです。

米国は，エコーキャンセラー方式と呼ぶ0～80kHzのISDN方式で，ADSLとはほとんど重ならない周波数を利用しています。これに対し日本国内では，0～320kHzを使うピンポン方式ISDNのため，**第１図**のようにADSLが占める周波数帯域と重なってしまい，相互に干渉し合い通常の通信が行えなくなります。

一昨年，国内でADSL接続サービスが初めて開始された当初，引き込まれている電話回線の状況で，サービスエリア内にもかかわらず，ADSLが利用できないといったトラブルがあったのは，このような理由が最大の原因です。

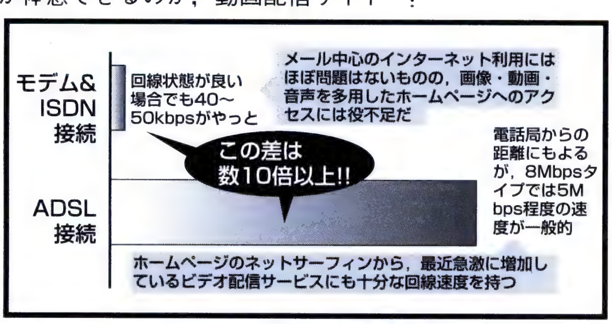
その対策をほどこしたのが，日本仕様のANNEX Cなのです。ANNEX Cは，ISDNのデータ送受信のタイミングに合わせ通信を行うデュアル・ビットマップという特殊な技術をもとにした特殊なADSL通信方式を採用することで，ISDNからの影響を可能な限り少なくしています。

FAQ 「ADSLでどこまで速くなるの？」接続スピードの実測値テスト

昨年末のADSL開通からほぼ１か月間，毎日のように通信速度テストを繰り返してきたが，やはりADSL 8 Mbpsの威力は絶大で，時間帯で差がつくものの，平均で5 Mbps前後の速度をマークしている。この数値は，アナログ回線モデムやISDNでの実測値の100倍近くにもなる。このスピードが体感できるのが，動画配信サイトなどで楽しめるビデオ・ストリーミング・サービスだ。ディスプレイのフル画面でも，多少MPEG-2特有のブロック・ノイズは気になるものの，十分に鑑賞できるレベル。また，数10MBもあるゲームがファイルも，分単位でダウンロードできるのも凄い!!　ネットサーファーには，ADSLは欠かせないはずだ。

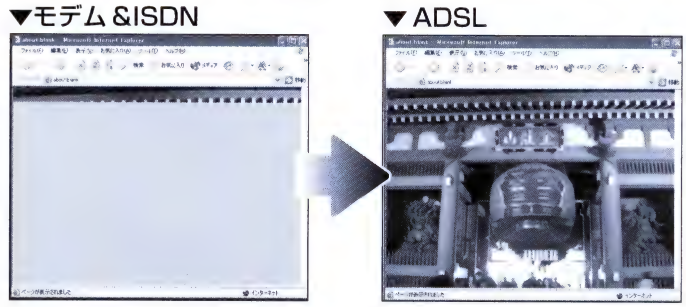


▲Windows Media Playerでのフル画面を使ったストリーミング・ビデオ再生も，満足できるスピードだ



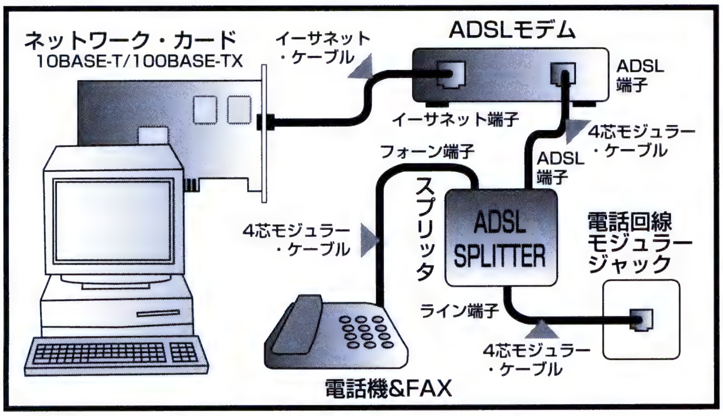
▲ADSLとISDN・モデム接続の速度差

▼モデム＆ISDN　　▼ADSL



モデム・ISDNとADSL回線の速度差を画像ダウンロードでイメージしてみると，このような違いになる。「ズルズル」だった描画がADSLでは「パッ・パッ!」と表示するように変わるといえば，わかりやすいだろう

❶ ダイレクト接続▶低予算でのADSL導入にベスト!!

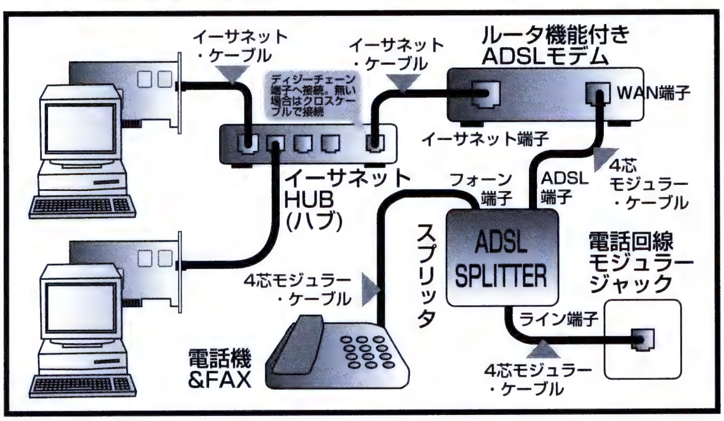


PC 1台のADSL接続の基本ともなる最も簡単な構成。パソコンにネットワーク端子があれば，ADSLモデム，スプリッター以外の機材は不要。導入初期費用をできるだけ安上がりにという人向け。

○ 何といってもハードウエアにかかる初期費用が安いこと。PC×1台なら，この構成からADSLへ入門するのもよいだろう。

× 複数PCの接続は無理。ADSLとPCがダイレクトに繋がっていることから，不正アクセスに弱いのが欠点といえよう。

❷ ルータ内蔵ADSLモデム接続▶HUB追加で複数PCで接続

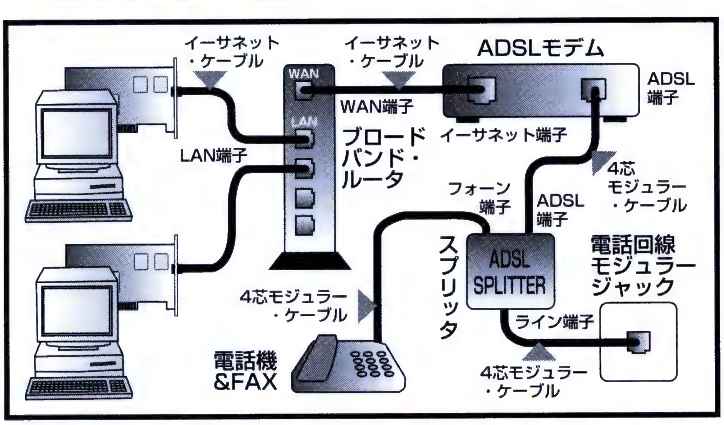


ルータ機能搭載のADSLモデムを用意するADSL業者もある。モデムとしてPC×1台からイーサネットHUB（4ポート・タイプなら4,000円前後）で複数のパソコンまでPC環境に合わせた構成が選べる。

○ 契約するADSL業者がルータ機能付きモデムをレンタルしている場合は低予算で，複数のPC環境にできる。

× ルータ部が簡易機能に限定され，ブロードバンド・ルータのような高度な設定はできない。HUB購入費も気になる

❸ モデム＆ルータ接続▶割高だか拡張性は良さがウリ!!



将来に向けても，まったく安心できるADSL接続のフル環境。ルータは多少高価だが，外部からの不正ハッキングを防ぐファイア・ウォール機能もあり，セキュリティ性は高い。予算があれば是非お勧め!!

○ セキュリティ対策ができる，HUBの追加でさらに接続PCの台数を簡単に増やせるなど，拡張性は高い

× 8Mbps対応の高速なブロードバンド・ルータは高価だが，ローエンド・モデルも充実してきたので，最近はそれほどではない

ISDNに日本独自の方式を選んだに仕方なく考案された規格ともいってよいでしょう。なお，「Yahoo!BB」のように，国内でもANNEX Aを採用したADSLサービスもあります。

ADSL接続に使われるPPPoE,PPPoAプロトコル

ADSL回線に関わるもうひとつの重要な用語が，「PPPoE」と「PPPoA」です。こちらは，インターネットへのアクセスに欠かせないネットワークの接続方式の規格です（**第2図**）。ネットワーク・プロトコルの一つであるこれらの規格は，プロバイダーと加入者のパソコンをADSL回線を通して1対1で接続するためのものです。通常，インターネットのWAN（ワイドエリア・ネットワーク）と加入者側の複数のパソコンでのLAN（ローカルエリア・ネットワーク）環境とを1対1通信であるADSL回線で接続することはできません。WANとLAN，ADSL回線を通過するデータ形式が異なっていると思ってもらえばよいでしょう。

これを変換するのが，PPPoE（ポイントツーポイント・プロトコル・オーバー・イーサネット）やPPPoA（〜・ATM）です。なお，ATMは非同期転送モードの略です。PPPoEとPPPoAは，ADSL接続サービスを行っているプロバイダーや利用する電話局の通信機器ごとに異なり，導入するサービスが指定する方式を利用することになります。

レンタルではなく市販のモデムやルータを購入する場合は，導入予定のADSLサービスの通信方式やプロトコルなどを事前に調べ，その規格に合った製品を購入しなければなりません。プロトコルの選択は，OSとの連携やモデム，ルータとの設定時に深く関わる項目なので，次回の実践セットアップ・テクニックでもう一度詳しく紹介します。

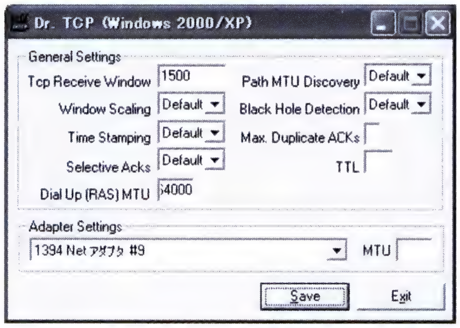
三つの方法から選べるADSLハードウエア設定

スプリッターやADSLモデム，ブロードバンド・ルータなど，ADSL接続に特

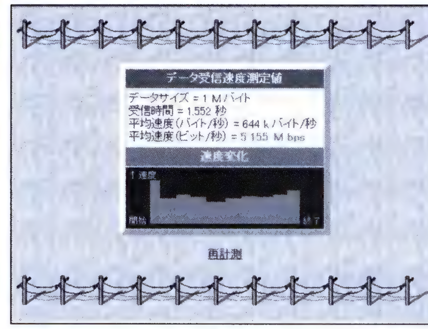
ユーティリティ・ソフトで最高速度をゲット!!

ADSLチューンナップ・テクニック

思い通りの速度が得られない場合にお勧めなのが，ADSLのチューンナップ。ADSLで用いる接続プロトコル「TCP」でデータ送受信を行う場合のパラメータ値「MTU」や「RWIN」を専用ソフトでOSや回線別の最適値に変更するだけで，速度のアップをはかる仕掛けだが，数割程度も向上する場合もある。最新のWindows XPでは，チューン済みのため効果は少ないが，Windows Meや98などを使っている人は試してみてはいかがだろう。



▲「RWIN」，「MTU」値変更ツールの代表が，「Dr.TCP」。利用には高度な知識が必要で初心者にはとっつきにくいものの，インターネット検索で探せば，設定方法を解説した数多くのサイトが見つかるので，それを参考にすればよい



▲変更ツールでの効果を試すには，回線速度の実測値が測れるベンチマーク・サイトが欠かせない。いろんなサイトが用意されているので，検索エンジンなどで探してみよう（写真はOSO氏提供のモデム・テストサイト）

有な機材があることは，すでに先月号で紹介したとおりですが，ここではそれらの機材を接続し，ADSLを導入するための方法を解説しましょう。

まず，ADSLをどのような環境で利用するのか，また予算がいくらまでなら許されるのか，考えてみてください。

一般的なADSL導入の構成を前ページに①～③まで挙げてみました。費用は①が最も安く，③は中級クラス向けのデラックス版にあたります。

パソコン1台だけの環境でADSLで高速インターネットをという場合は，お手軽モデルの①に決まりです。もちろん，ADSLモデムとパソコン間はネットワーク接続を行うため，ネットワーク端子のないパソコンは，LANカード（10Base-T／100Base-TXタイプ）の増設が必要です。このカードは安い製品であれば，2,000円前後で購入できるので，モデムとスプリッターをリースすれば，お小遣いの範囲で用意できます。もちろん，毎月の通信費用は別ですが，それでも4,000円程度なので，安心ですね。心配な点は，直結することでセキュリティに難ありといったことでしょうか。

②と③は，複数のパソコンがある環境での構成モデルです。ADSLサービス業者のなかには，リースしているモデムにブリッジ・モード（単なるモデム機能だけ）のほかに，ルータ機能を持つモデムを選べる場合もあります。このようなルータ内蔵モデムの場合，イーサネットHUB（ハブ）というLAN用の配線分岐ボックスをつなぐことで，複数のパソコンを接続し，各機から同時にインターネットへ接続することも可能です。

ルータ内蔵モデムは，通常のADSLモデムよりは，リース料金が多少高く設定されているようです。とはいえ，③のように本格的なブロードバンド・ルータを購入するよりは導入費用は少なくて済みます。

デラックス版の③モデルは，予算さえ許せばぜひ実現していただきたい，ADSL環境です。最近では高速な通信性能を持つブロードバンド・ルータも発売されはじめ，これまでネックとなっていた8 Mbpsサービス導入時の速度低下も，それほど気にならなくなっています。嬉しいことに，1万円以下のモデルが続々登場していますので，①②モデルとの導入費用の差は以前よりはなくなっています。パソコン1台からもルータを利用可能で，予算に多少余裕があるようなら，将来のことも考えて最初から③モデルを実現してみてはいかがでしょう。

超快適なアクセスがこれだけの料金でいいの!!

最初の約束どおり，ASDL会社から送られてきた，ADSLの通信料金を大公開してしまいましょう。なお，自宅に引いたADSLは，8 Mbps料金では国内（世界？）最安値のあの「Yahoo!BB」です。

●Yahoo!BBからの1月分請求書内訳

<月額費用>

NTT回線使用料（タイプ1）：187円
ADSLサービス：990円
ISPサービス：1,290円
ADSLモデム・スプリッタレンタル料：550円

<初期費用>

NTT局内工事料金（ISDN）：2,800円
（ISDNからアナログ回線への変更）
NTT DSL契約料：800円

それに，NTT東日本からの電話回線使用料の2,514円が加わります。

このうち，ADSL導入時にかかる初期費用を除けば，導入後のADSL分の毎月の支払いは3,000円ほどで，電話代も含めても5,000円台で済みました。これまで利用していたISDN常時接続サービス（フレッツISDN）では，プロバイダー接続料金と合計すると，1万円近くも払っていたのですから，約半分の節約になります。しかも，64kbpsに比較にならないほどの高速性能を考えると，ISDNにこれまで支払ってきた金額は…と悩んでしまうほどの激安料金です。

（文・福場龍夫）

マンガで学ぶ コンピュータの世界

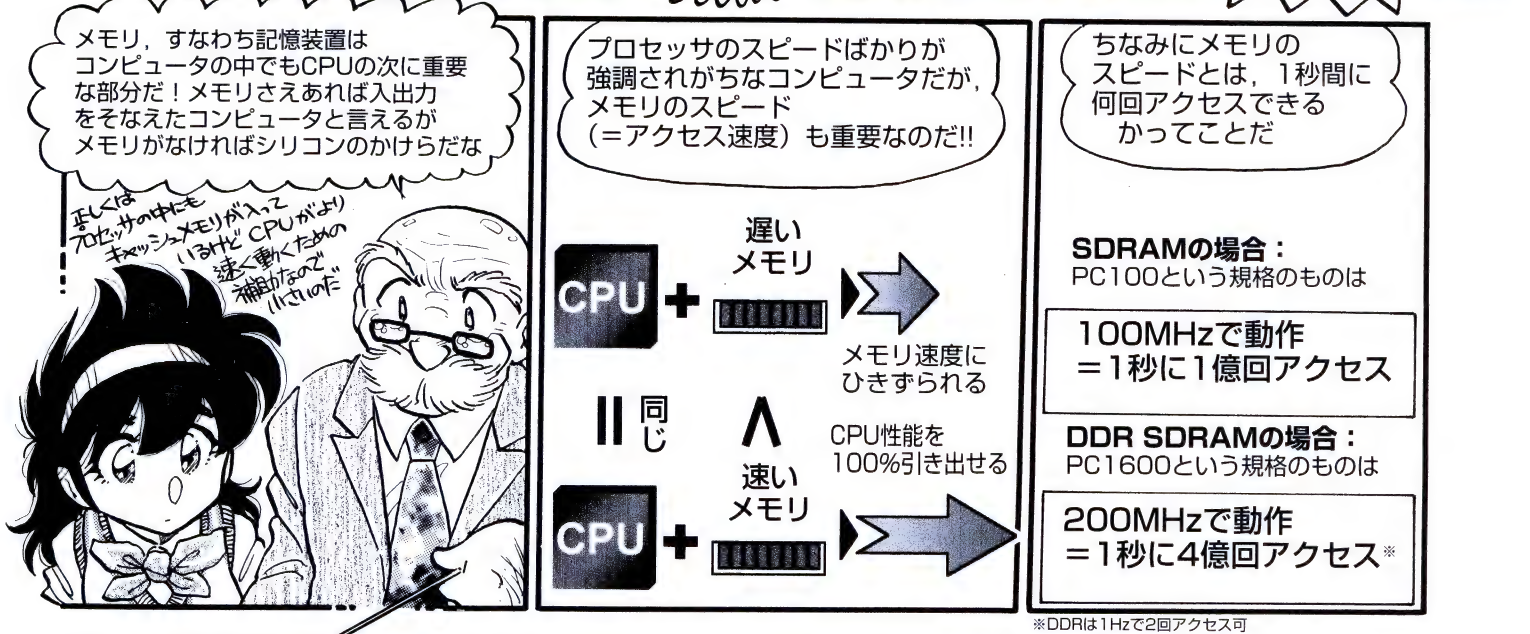
パソコンレクチャー ザ・ビギニング

「忘れちゃいけないメモリ！」

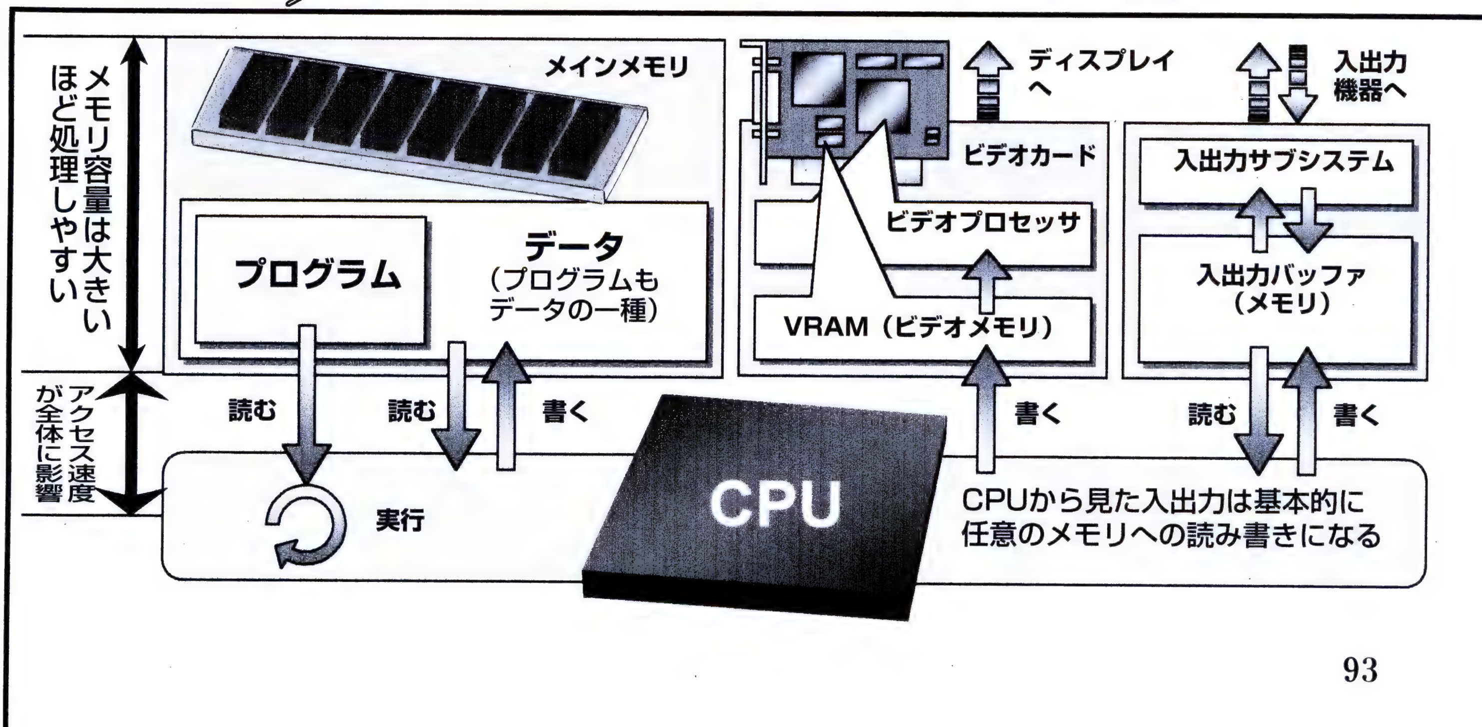
by くりひろし



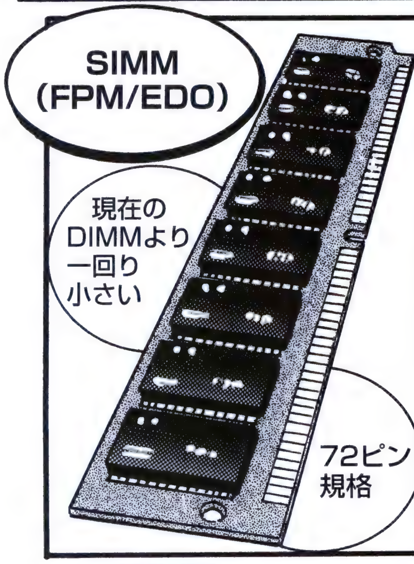




※DDRは1Hzで2回アクセス可

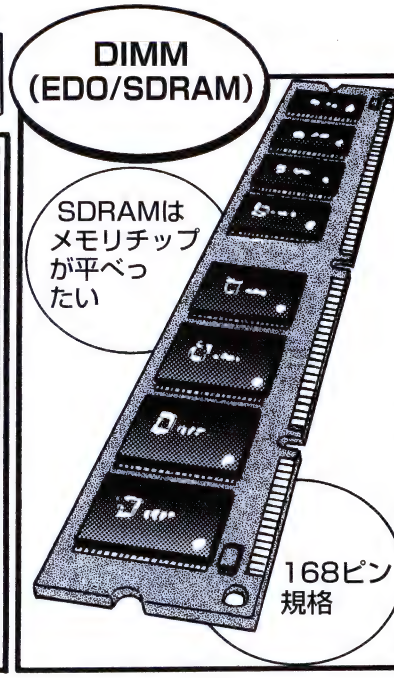


パソコン用メモリ・モジュールいろいろ



Single In-line MemoryModule

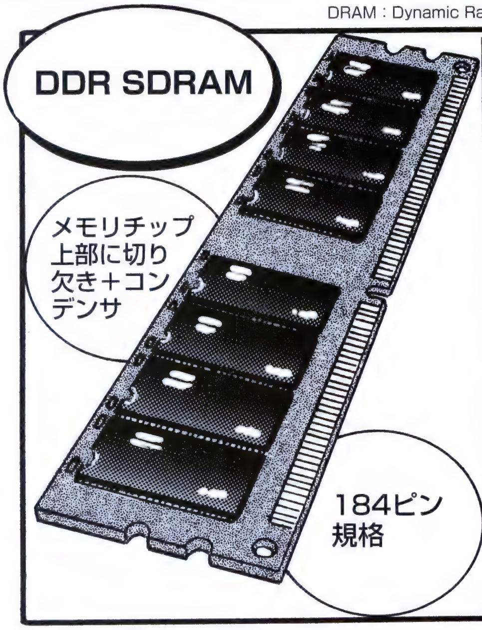
初期の頃はFPM（Fast Page Mode）DRAMをメモリチップに使っていたが95年位からより高速のEDO（Exteded Date Out）型になった。Pentiumプロセッサと使用する場合,必ず2枚1組で使う。97～98年頃まで使われたが現在はない。



Dual In-line Memory Module

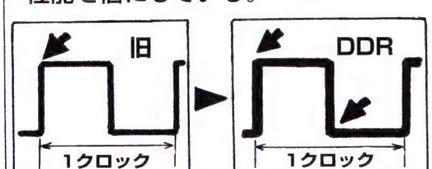
接続端子が両面に（Dualに）並んでいるのでPentiumに対しても1枚ずつ使用できる。初期はEDO DRAMだったが,現在はSDRAM（Synchronous DRAM）をメモリチップに使っている。SDRAMは外部動作クロック周波数に同期してアクセスすることで高速化した。今のところ数も多く使われているメモリだ。

DRAM：Dynamic Random Access Memory



Double Data Rate

基本的にはDIMMであり，SDRAMを使っていることも同じだが，より速く動作させるためにノーマルのSDRAMが1クロックで1回アクセスするのに対し2回アクセスさせて性能を倍にしている。



技術的にも従来のSDRAM同様であるので，価格も安く作れる。当初Athlonしか対応してなかった※が，今やPentium用もあり，今後の主流。

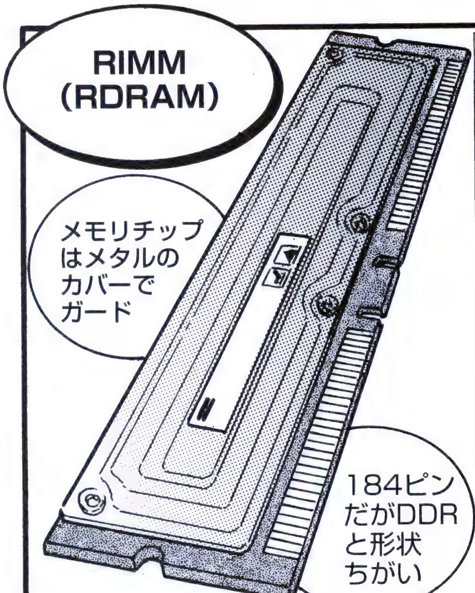
※正しくはマザーボードが対応してなかった。



- 対応するチップセット増加中
- 一般に同じレベルSDRAMの2倍の性能
- RDRAMよりずっと安価

⇕

- より高速，最速メモリ
- 対応チップセットが限られる
- まだコストがかかる



RDRAM In-line Memory Module

メモリチップにRDRAM（Rumbus DRAM）を採用し，高速化・大容量化が期待できる次世代メモリ。
RDRAMでは外部とのインターフェースにダイレクトRambus方式を使い，DDRよりも高速に動作できるが，対応しているマザーボードが少なく，価格も高かったため，パソコンのメインメモリの主流になるには時間がかかりそう。
一方，高速性を生かしてビデオカードのVRAMや，PS2，N64などの家庭用ゲーム機にも使われている。





メモリの大容量化はその時代のOSにあわせて育ってきた。現在の推奨メモリは
Win95 快適32MB
Win98 快適64MB
WinXP 快適256MB
256MBくらいじゃが…ビデオ編集や数百万画素のフォトレタッチを考えると512MBくらいはほしいの
さて今後のメモリじゃが…DDR系の主流はつづくじゃろうが
小型化
DIMM
SO-DIMM
DDR 2000年～ 100～300MHz
DDR-Ⅱ 2002年～ 400～500MHz
DDR-Ⅲ 2004年～ 600～800MHz
ADT Advanced DRAM Technology
統合？
限界があるのでいずれはRIMMに移行するかな
RIMM
より高速化，大容量化
RIMM 4200 (32bit RIMM)
ところでワシが心配なのは円安とテロのせいで，メモリ価格がアップしてきとるんじゃよなー…
メモリ価格 変動状況
2001
2002
そうそう！忘れないでねっホワイトデーのおかえしっ
え～～～？そもそも今年はバレンタインのチョコもらってなかったような…
ニコ ニコ
あげたわよ。
え，あの，その，ね，Dr.Dも何ももらってないよね？
さぁー…朝メシも忘れるような年寄りに聞かれてもなぁ？
えええ だって，あの その
くどいっ
あれ…？もらって…
たかも……しれない？気がする…
ニヤリ
勝った！
TO BE CONTINUED...
はたして前年のコトを覚えてるかどうか…？

もっと身近にプログラミング

チャレンジ!!

オリジナル・プログラム言語の制作！

その2 変数システムと四則演算

文 まかべひろし

先月まで記事を書かせてもらっていた「P/ECE」でのプログラミングにハマる毎日です。自分のプログラムが持ち歩けるのもそうですけど，デバイスの割にやたら早いCPUがプログラミングをさらに楽しませてくれてます。

絵 桜部のとす

D-VHSのビデオデッキ買いました～。もう今まで使ってたデッキが寿命っぽかったので買い換えって事で。や，実際キレイですね。ホントにオンエアと変わらないレベル。今までに撮ったテープもキレイに再生してくれるのでかなり嬉しいですよ～。

catlikeホームページ ▶ http://catlike.marchen.to/

ロールプレイング・ゲームやアドベンチャー・ゲームでは必ずお世話になるスクリプト，さらに発展させてオリジナル言語を作っちゃおう！ 今回はプログラムの基礎，変数部分を実装するよ！

2月22日にちょっと期待

ついにMicrosoftのゲーム機Xboxが発売です！海外産のゲーム機は日本で軒並み倒れた過去を振り切って，巨人の放つでかい箱は成功するのでしょうか。個人的には，同時発売の「ねずみくす」のふさふさ感が気になって気になってしょーがないのです。

ただ，やっぱり一番気になるのは，個人レベルで開発できるのかどーか。ううむ，ブートさえできれば，ハードウェアは扱いなれたWindowsっぽいんだけどなぁ・・。

オリジナル言語，どこから手をつける

さぁて。今回からいよいよ「オリジナル言語」制作にとりかかっていきましょうか。と，その前に，前回決めた仕様の確認確認。ここで言うオリジナル言語は，メモ帳やテキストエディタで記述した「マクロファイル」を解析しながら，その内容に沿ってプログラムを実行していく形になります。

前回の最後に，言語システムのベースとなるプログラムを紹介しましたが，こちら。

リスト1では，macro.txtというテキストファイルをメモリに読み込み(A)，ファイルの先頭から1文字ずつ見ていけるようにループを組んでいます(B)。ループ内でIsKanji()というルーチ

≪リスト1≫

```
#include <stdio.h>
#include <stdlib.h>

// 漢字コード判定用
int IsKanji( unsigned char cData ) {
        if ( cData < 0x81 ) { return 0; }
        if ( cData < 0xa0 ) { return 1; }
        if ( cData < 0xe0 ) { return 0; }
        if ( cData < 0xff ) { return 1; }
        return 0;
}

void main() {
        FILE *fp; int nSize, nPnt;
        unsigned char *p;
        fp = fopen( "macro.txt", "rb" );                    //<<<<< (A)
        fseek( fp, 0, SEEK_END ); nSize = ftell( fp );
        fseek( fp, 0, SEEK_SET );
        p = ( unsigned char* )malloc( nSize + 1 );
        fread( p, 1, nSize, fp );
        fclose( fp );
        nPnt = 0;
        while( nPnt < nSize ) {                                      //<<<<< (B)
                if ( IsKanji( p[nPnt] ) ) {
                        nPnt += 2;
                } else {
                                                                     //<<<<< (C)
                        nPnt++;
                }
        }
        free( p );
}
```

ンを呼んでいますが，これは，今見ている文字がSHIFT-JISにおいて全角文字かどうかを判断するルーチンです。マクロファイルは全角・半角が入り混じっているかもしれません。もし全角文字なら，1文字あたり2byte進める必要がありますよね。

また，プログラム言語は半角文字で書くのが通例です。全角文字は命令として機能しないのは，普段プログラムを書いているみなさんなら当たり前のことだと思います。そこで，全角文字がもし来てしまった場合，そのまま何もせず2byte読み進めることにします。命令の処理を行うのは，半角文字が来た場合のみとすればいいわけですね。

ではでは，**リスト1**をベースに，言語としての機能を追加していきましょう。

とは言っても，さて，なんの機能を追加しましょうか。こういう時はまず，仕様を見直すと，やるべきことが見えてくるものです。

ベーマガ(本誌)マクロプロセッサ(仮)
略称「BMP」仕様草案

- 整数値を格納できる変数システム
- 演算処理は四則演算対応
- 条件分岐にIF文。もちろん入れ子対応
- BASICライクにGOTO命令でジャンプ
- サブルーチン処理用にGOSUB/RETURN
- 文字列変数や配列も欲しいところ
- デバッグ用に，文字列を画面出力
- マクロ内にコメントを書けるようにする

プログラム言語と言って，一番最初に覚えたことは何だったかな？もうずいぶん昔で覚えてませんか？だいたい，どの教科書を見ても，最初にやることは画面に文字や映像を表示することのようです。打ち込んだ内容がそのまま画面に反映される。プログラム初心者にとってそれが一番興味を引くことであるのは間違いないでしょう。

しかし，BMPの仕様草案には画像まわりは入っていません。忘れた？いえいえ。画像や音，キーボードなど入出力を扱う部分というのはプラットホームによって環境はさまざまです。ベーマガという雑誌はハードにとらわれず，さまざまなプラットホームを扱っているわけですから，BMPもプラットホームに依存するわけにはいきません。

さて，初心者の興味云々は関係なく，プログラムの基本である処理はなんでしょう。もちろん仕様草案にそれは盛り込んであります。草案の項目順番は，実は実装していく機能の順番でもあるのです。そう，プログラムと言えば，最初はやっぱり「変数」からですよね。条件判断するにも変数が無ければどうにもならないし，計算結果を保存しておく場所も欲しいものです。

今回のプログラム(**リスト3**)は結構長く複雑ですが，各機能の解説とあわせながら見ていくとわかりやすいかと思います。

≪リスト2≫

```
:PRINT 1+2*3
:$HENSU = 8
:PRINT $HENSU
:PRINT $HENSU * 2
```

また，**リスト2**のようなマクロファイルを読ませて解析できるようにしたいと思います。

変数の仕様と実装を考える

では，BASICやC言語の場合，変数はどのような書式で使うでしょうか。一番基本となるのは，変数への代入です。BASICだとこんな感じですよね。

```
HENSU = 10
```

左側に変数名，イコールでつないで，右側に代入する値，という見慣れた形です。BMPでは，プロセッサ側の処理を簡単にするために，命令の頭には必ず「:」をつけることにしたいと思います。変数代入も命令には違いませんから，頭には「:」をつけます。

```
:HENSU = 10
```

これでOK？　いやいや，まだ問題があります。このままでは，IFやGOなどの命令を今後つけた場合，変数名としてIFやGOという変数は使えないことになります。他の言語はそれでも問題なかったりしますが，よりプロセッサ側の処理を簡単・明確にするために，変数の場合は頭に「$」をつけることにしてみましょう。

```
:$HENSU = 10
```

このようにすると，「:」の後に「$」が来た場合は変数代入，というのが明確にわかると思います。**リスト1**に加えるならば，(C)の位置に，

```
if ( p[nPnt] == ':' ) {
        nPnt++;
        if ( p[nPnt] == '$' ) {
                // 変数代入
                // 変数名を得るサブルーチンを呼ぶ
                // 「=」の次まで飛ばす
                // 代入すべき値を得る
                // 変数に値を代入する
        } else {
                // 命令はこちら・・
        }
}
```

こんな感じで判定を追加します。後は，$の後に続く「=」までの文字列を変数名と判定すればOKですね。ここを工夫すれば，日本語の変数というのも処理可能だったりするのですが，プログラム処理に日本語が混ざると記述ミス等の問題がおきやすいので，とりあえず変数で使える文字は半角英数文字のみとしておきましょう。

ここまで決まれば変数回りの実装を行うことになります。変数は基本的に，変数への書き込みと，変数からの読み出しができれば十分ですので，それらに必要なプログラムをサブルーチンとして用意すればよいのです。

変数は基本的に，メモリ内に配置された「変数名」と「値」の対になったデータです。これは構造体や配列変数を使うことで実装することができま

```
typedef  struct {
        char        szName[32];
        int                 n;
} VALUESTRUCT;
```

≪第1図≫ 変数の構造体定義

す。BMPでは，変数名は32byte以内，値は整数のみと考え，**第1図**のように構造体を作成しました。

この構造体定義を配列として扱い，作った変数の分だけメモリ内に確保してやればいいわけです。新しい変数の作成は，変数へ値を代入するとき，同じ変数名で作られたものが無ければ新たに作成する，という手順で行います。

変数システム用ルーチン群

初期化:	void ValueSysInit()
解放:	void ValueSysRelease()
文字列から変数名取得:	void ValueSysGetValueName(文字列, 変数名バッファ, バッファサイズ)
変数名検索:	int ValueSysSearchValue(変数名)
変数に値を代入:	void ValueSysSetData(変数名, 値)
変数から値を得る:	int ValueSysGetData(変数名)

四則演算ルーチンを実装しよう！

変数自体に代入するための仕組みはできましたが，実際は，算式の右辺にあたる部分の解析も行わなければ変数代入を実現することができません。前述の例のように：$HENSU = 10であれば，文字列で書かれている「10」の部分を数値の「10」と認識しなくては，せっかく用意した変数システムも無用の長物になってしまいます。

また，変数に代入するのは固定数値だけではありません。

:$HENSU = 1+2*3

のような計算式，もちろん計算式の中に

:$HENSU = $HENSU * 2

みたいに変数が入っていても，正しく認識できる必要があるでしょう。

さて，数値計算自体は，「逆ポーランド記法」という考え方を用いることで実装することにします。この方法は，1+2*3のように，四則演算の順番によって後ろの方から計算しなくてはいけない場合も，正しく計算することができるので非常に便利です。

普段ぼくらが使っている算術式は「1+2*3」のように，数値と数値の間に演算記号を挟み込む形で記述していますが，この逆ポーランド記法は「1 2 3 * +」といった具合に記述します。計算するときは，「記号から左ふたつの数値を用いる」というルールにのっとって処理します。「1+2*3」と違い，必ず左側から解析していけば良いのがプログラム側からするととても都合がいいのです。

しかし，いくら便利であるからといって，普段なじみの無い算式をオリジナル言語で書かせるわけにはいきません。そこで，一度逆ポーランド記法に変換し，その後計算という段階を踏むことにします。詳しくは，本誌2000年10月号の「catlikeのもっと身近にプログラミング！」で取り上げていますので，そちらをご覧下さい。

以前紹介した計算ルーチンでは変数に対応していませんでしたが，今回は「$HENSU * 2」を実現させるためにも，変数システムに対応させなくてはいけません。ちょっと難しそうに思えますが，これは計算式の各項を得てくるとき，「$」から始まっていたらそこからを変数とみなすと考えれば簡単です。これは，**リスト3**のGetIntSingleFromMemory()で実装しています。

この四則演算ルーチンができあがってしまえば，変数代入はできたも同然。右辺の取得に四則演算ルーチンを用いれば，計算式であろうが，ただの変数から変数への代入であろうが，最終的にはひとつの「値」になり，変数代入を行うことができます。もちろん，今後「命令」を追加した場合，そのパラメータ取得にも使うことができるわけです。

四則演算用ルーチン

計算をする：int GetIntFromString(計算式文字列, 結果を格納する場所);

動きを確認してみよう！

変数計算部を実装したものが**リスト3**です。このプログラムは，macro.txtというテキストファイル(**リスト2**)を読み込んで動作するので，そちらも用意しておきましょう。デバッグ用途に:PRINTという命令を用意しておきました。これは，次に続く算術式の答えを画面に出力する命令です。さすがに，画面に何も出ないと，ちゃんと動いているのかどうかもわかりませんもんね。



▲コンソールで実行した結果

≪リスト3≫

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

// 変数系処理 //////////////////////////////////////////////////////////////////////

typedef     struct {
                char        szName[32];
                int             n;
} VALUESTRUCT;
VALUESTRUCT *pValue = NULL;
int nValueNum;
int nValueMaxNum;

// 変数システムの初期化
void ValueSysInit() {
        if ( !pValue ) {
                nValueNum = 0;
                nValueMaxNum = 32;
                pValue = (VALUESTRUCT*)malloc( sizeof(VALUESTRUCT) * nValueMaxNum );
        }
}

// 変数システムの解放
void ValueSysRelease() {
    if ( !pValue ) {
            free( pValue ); pValue = NULL;
    }
}

// 文字列から変数名を得る
int ValueSysGetValueName( char *pStr, char *pBuf, int nBufSize ) {
    int i = 0;
    while( i < nBufSize ) {
```

→つぎのページに続く

```
			if ( ( 'A' <= pStr[i] && pStr[i] <= 'Z' )
					|| ( 'a' <= pStr[i] && pStr[i] <= 'z' )
					|| ( '0' <= pStr[i] && pStr[i] <= '9' )
			|| ( pStr[i] == '_' ) ) {
					pBuf[i] = pStr[i];
			} else {
					break;
			}
			i++;
	}
	pBuf[i] = '¥0';
	return i;
}

// その変数が実在するか確認
int ValueSysSearchValue( char *szName ) {
		int i;
		for ( i = 0; i < nValueNum; i++ ) {
			if ( strcmp( szName, pValue[i].szName ) == 0 ) { return i; }
		}
		return -1;
}

// 変数に値を代入する
void ValueSysSetData( char *szName, int nData ) {
	int n = ValueSysSearchValue( szName );
	if ( n == -1 ) {
			if ( nValueMaxNum == nValueNum ) {
					nValueMaxNum *= 2;
					pValue = (VALUESTRUCT*) realloc( pValue,sizeof(VALUESTRUCT)
* nValueMaxNum );
			}
			strcpy( pValue[nValueNum].szName, szName );
			pValue[nValueNum].n = nData;
			nValueNum++;
	} else {
			pValue[n].n = nData;
		}
}

// 変数の値を得る
int ValueSysGetData( char *szName ) {
		int n = ValueSysSearchValue( szName );
		if ( n == -1 ) {
				return 0;
		}
		return pValue[n].n;
}

// 逆ポーランドを用いた算出関数（小数には対応していません）
////////////////////////////////////

unsigned long g_dwStackCnt = 0;
int g_pStackMem[ 128 * 2 ];				// 128項まで対応

// スタックにデータをPush
int StackPush( int nData ) {
		g_pStackMem[ g_dwStackCnt ] = nData;
		g_dwStackCnt++;
		return 1;
}

// スタックからデータをPop
// データが無ければ0
int StackPop( int *nData ) {
		if ( g_dwStackCnt == 0 ) { return 0; }
		g_dwStackCnt--;
		*nData = g_pStackMem[ g_dwStackCnt ];
		return 1;
}

// 計算関連

// スペースやコントロールコード等を飛ばす
unsigned long SkipSpaceFromMemory( unsigned char *pSrc ) {
		unsigned long dwCnt = 0;
		while( pSrc[dwCnt] < ' '+1 ) { dwCnt++; }
		return dwCnt;
}

// メモリから１つの数値を得る(計算しない)
unsigned long GetIntSingleFromMemory( unsigned char *pMath, int *iAnswer ) {
		unsigned long dwCnt = 0;
		int iSubFlg = 1, iData = 0;
		int iCode = 10;			// 進数
		// スペースやTABを飛ばす
		dwCnt += SkipSpaceFromMemory( &pMath[dwCnt] );
		switch( pMath[dwCnt] ) {
			case '+':
					dwCnt++; break;
			case '-':
					iSubFlg = -1; dwCnt++; break;
		}
		// 変数対応処理
		if ( pMath[dwCnt] == '$' ) {
				char szValueName[32];
				dwCnt++;
				dwCnt += ValueSysGetValueName( (char*)&pMath[dwCnt],
szValueName, sizeof(szValueName) );
				iData = ValueSysGetData( szValueName );
		} else {
				while( '0' <= pMath[dwCnt] && pMath[dwCnt] <= '9' ) {
						iData = iData * 10 + pMath[dwCnt] - '0';
						dwCnt++;
				}
		}
		// 戻り値作成
		*iAnswer = ( iData * iSubFlg );
		return dwCnt;
}

// プライオリティ番号
#define	REPAULAND_PRIO_OPEN			10
#define	REPAULAND_PRIO_NUM			9
#define	REPAULAND_PRIO_NOT			8
#define	REPAULAND_PRIO_DOUBLE		7
#define	REPAULAND_PRIO_MUL_DIV		6
#define	REPAULAND_PRIO_ADD_SUB		5
#define	REPAULAND_PRIO_PAU_AND_OR4
#define	REPAULAND_PRIO_Q_EQU_NOT 3
#define	REPAULAND_PRIO_Q_AND_OR		2
#define	REPAULAND_PRIO_CLOSE		1
#define	REPAULAND_PRIO_NULL			0

// 記号番号
#define	REPAULAND_TYPE_EQU_EQU		19		// ==
#define	REPAULAND_TYPE_NOT_EQU		18		// !=
#define	REPAULAND_TYPE_OVER_EQU		17		// >=
#define	REPAULAND_TYPE_MIN_EQU		16		// <=
#define	REPAULAND_TYPE_AND_AND		15		// &&
#define	REPAULAND_TYPE_OR_OR		14		// ||
ここまで2文字記号
#define	REPAULAND_TYPE_OVER			13
// >
#define	REPAULAND_TYPE_MIN			12
// <
#define	REPAULAND_TYPE_OPEN			11
// (
#define	REPAULAND_TYPE_NOT			10
// !
#define	REPAULAND_TYPE_DOUBLE		9		// ^
#define	REPAULAND_TYPE_MUL			8
// *
#define	REPAULAND_TYPE_DIV			7
// /
#define	REPAULAND_TYPE_ADD			6
// +
#define	REPAULAND_TYPE_SUB			5
// -
#define	REPAULAND_TYPE_PAU			4
// %
#define	REPAULAND_TYPE_AND			3
// &
#define	REPAULAND_TYPE_OR			2
// |
#define	REPAULAND_TYPE_CLOSE		1		// )
#define	REPAULAND_TYPE_NUM			0
// 0 ～ 9

// 逆ポーランドにおける優先順位を得る
int GetRevPaulandPrio( unsigned char *c, int *iTypeNum ) {
		// 項目の優先順位を得る
		int iPrio = REPAULAND_PRIO_NULL;
		if				( c[0] == '=' && c[1] == '=' )
					{ iPrio = REPAULAND_PRIO_Q_EQU_NOT; *iTypeNum =
REPAULAND_TYPE_EQU_EQU; }
		else if ( c[0] == '!' && c[1] == '=' )
					{ iPrio = REPAULAND_PRIO_Q_EQU_NOT; *iTypeNum =
REPAULAND_TYPE_NOT_EQU; }
		else if ( c[0] == '>' && c[1] == '=' )
					{ iPrio = REPAULAND_PRIO_Q_EQU_NOT; *iTypeNum =
REPAULAND_TYPE_OVER_EQU; }
		else if ( c[0] == '<' && c[1] == '=' )
					{ iPrio = REPAULAND_PRIO_Q_EQU_NOT; *iTypeNum =
REPAULAND_TYPE_MIN_EQU; }
		else if ( c[0] == '>' )
					{ iPrio = REPAULAND_PRIO_Q_EQU_NOT; *iTypeNum =
REPAULAND_TYPE_OVER;	}
		else if ( c[0] == '<' )
					{ iPrio = REPAULAND_PRIO_Q_EQU_NOT; *iTypeNum =
```

→ つぎのページに続く

```
REPAULAND_TYPE_MIN;    }
                else if ( c[0] == '&' && c[1] == '&' )
                                { iPrio = REPAULAND_PRIO_Q_AND_OR;   *iTypeNum =
REPAULAND_TYPE_AND_AND; }
                else if ( c[0] == '|' && c[1] == '|' )
                                { iPrio = REPAULAND_PRIO_Q_AND_OR;   *iTypeNum =
REPAULAND_TYPE_OR_OR;   }
                else if ( c[0] == '(' )
                                { iPrio = REPAULAND_PRIO_OPEN;      *iTypeNum =
REPAULAND_TYPE_OPEN;    }
                // 変数対応処理
                else if ( ( '0' <= c[0] && c[0] <= '9' ) || ( c[0] == '$' ) )
                                { iPrio = REPAULAND_PRIO_NUM;       *iTypeNum =
REPAULAND_TYPE_NUM;     }
                else if ( c[0] == '!' )
                                { iPrio = REPAULAND_PRIO_NOT;       *iTypeNum =
REPAULAND_TYPE_NOT;     }
                else if ( c[0] == '^' )
                                { iPrio = REPAULAND_PRIO_DOUBLE;    *iTypeNum =
REPAULAND_TYPE_DOUBLE;  }
                else if ( c[0] == '*' )
                                { iPrio = REPAULAND_PRIO_MUL_DIV;   *iTypeNum =
REPAULAND_TYPE_MUL;     }
                else if ( c[0] == '/' )
                                { iPrio = REPAULAND_PRIO_MUL_DIV;   *iTypeNum =
REPAULAND_TYPE_DIV;     }
                else if ( c[0] == '+' )
                                { iPrio = REPAULAND_PRIO_ADD_SUB;   *iTypeNum =
REPAULAND_TYPE_ADD;     }
                else if ( c[0] == '-' )
                                { iPrio = REPAULAND_PRIO_ADD_SUB;   *iTypeNum =
REPAULAND_TYPE_SUB;     }
                else if ( c[0] == '%' )
                                { iPrio = REPAULAND_PRIO_PAU_AND_OR; *iTypeNum =
REPAULAND_TYPE_PAU;     }
                else if ( c[0] == '&' )
                                { iPrio = REPAULAND_PRIO_PAU_AND_OR; *iTypeNum =
REPAULAND_TYPE_AND;     }
                else if ( c[0] == '|' )
                                { iPrio = REPAULAND_PRIO_PAU_AND_OR; *iTypeNum =
REPAULAND_TYPE_OR;      }
                else if ( c[0] == ')' )
                                { iPrio = REPAULAND_PRIO_CLOSE;     *iTypeNum =
REPAULAND_TYPE_CLOSE;   }
                return iPrio;
}

unsigned long GetIntFromString_Sub( unsigned char *pMath,int *iAnswer ) {
                // 文字列を読み取り，逆ポーランドに変換
                unsigned long dwCnt = 0;
                // 逆ポーランドにした結果を入れる場所(最大1024項目)
                int *pAns = (int*)malloc( 1024 * sizeof( int ) * 2 );
                int iAnsCnt = 0;
                // 逆ポーランド算式の出力
                {
                                // ワーク用変数
                                int iStackTopType,iStackTopNum,iNum;
                                int iKigoFlg = 1;               // 前回が演算記号なら1
                                // 項目取得ループ
                                while( 1 ) {
                                                int iTypeNum, iPrio;
                                                // スペースやTABを飛ばす
                                                dwCnt += SkipSpaceFromMemory( &pMath[dwCnt] );
                                                // 項目の優先順位を得る
                                                iPrio = GetRevPaulandPrio( &pMath[dwCnt]
,&iTypeNum );
                                                // 項目無かった
                                                if ( iPrio == REPAULAND_PRIO_NULL ) { break; }
                                                // 前回が演算記号なのに今回 + or - だった
                                                if ( iKigoFlg && iPrio ==
REPAULAND_PRIO_ADD_SUB ) {
                                                                // マイナス数値の恐れあり.
                                                                iPrio = REPAULAND_PRIO_NUM;
                                                }
                                                // スタックの頭を取りだし，戻します。
                                                if ( StackPop( &iStackTopType ) ) {
                                                                // スタックにデータが入っていた
                                                                StackPop( &iStackTopNum );
                                                                StackPush( iStackTopNum );
StackPush( iStackTopType );
                                                } else {
                                                                // スタックにデータは入っていなかっ
た
                                                                iStackTopType =
REPAULAND_PRIO_NULL;
                                                }
                                                // スタックの頭と比べます
                                                if ( iStackTopType == REPAULAND_PRIO_OPEN &&
iPrio == REPAULAND_PRIO_CLOSE ) {
                                                        // スタックに ( が，参照記号に ) が来た
場合の特例.
                                                                // どちらも相殺され消える.
                                                                StackPop( &iStackTopType ); StackPop(
&iStackTopNum );
                                                                dwCnt++;
                                                } else if ( ( iStackTopType %
REPAULAND_PRIO_OPEN ) < iPrio || iStackTopType == REPAULAND_PRIO_NULL )
                                                                // スタックの頭より優先順位が高いので
積み上げます
                                                                if ( iPrio == REPAULAND_PRIO_NUM ) {
                                                                // 数値の場合，値を得る
                                                                                dwCnt +=
GetIntSingleFromMemory( &pMath[dwCnt], &iNum );
                                                                                iKigoFlg = 0;
                                                                } else {
                                                                                // 記号の場合，int 型
に変換しておく
                                                                                iNum = iTypeNum;
                                                                                // 2 文字系の記号か，
 1 文字系の記号か
                                                                                if ( iTypeNum <
REPAULAND_TYPE_OR_OR ) { dwCnt++; }

else { dwCnt += 2; }
                                                                                iKigoFlg = 1;
                                                                }
                                                                // スタックに優先順位とデータを積む
                                                                StackPush( iNum ); StackPush( iPrio );
                                                } else {
                                                                // スタックの頭の方が優先順位が高いので
，取り出して答えに追加します
                                                                int iType = 0;        // 種類. 0:数値,1:記
号
                                                                StackPop( &iStackTopType ); StackPop(
&iStackTopNum );
                                                                if ( iStackTopType !=
REPAULAND_PRIO_NUM ) {
                                                                                // 種類は記号である.
                                                                                iType = 1;
                                                                }
                                                                pAns[ iAnsCnt + 0 ] = iType;
                // 種類(プライオリティ)
                                                                pAns[ iAnsCnt + 1 ] = iStackTopNum;
// データ
                                                                iAnsCnt += 2;                     // 答え
カウンタを + 2
                                                }
                                }
                                // スタックにたまっているデータを出力
                                while( StackPop( &iStackTopType ) ) {
                                                int iType = 0;      // 種類. 0:数値,1:記号
                                                StackPop( &iStackTopNum );
                                                if ( iStackTopType != REPAULAND_PRIO_NUM ) {
                                                                // 種類は記号である.
                                                                iType = 1;
                                                }
                                                pAns[ iAnsCnt + 0 ] = iType;
// 種類(プライオリティ)
                                                pAns[ iAnsCnt + 1 ] = iStackTopNum;
// データ
                                                iAnsCnt += 2;
// 答えカウンタを + 2
                                }
                                // 種類に終端記号(-1)を付加して，この式を終了とする.
                                pAns[ iAnsCnt + 0 ] = -1;           // 種類(プライオリティ)
                                pAns[ iAnsCnt + 1 ] = 0;            // データ
                }               // of ここまでが逆ポーランド算式の出力

                ///////////////////////////////////////////////////////////////////
                // 逆ポーランド式の計算
                {
                                int i = 0;
                                while ( pAns[i] != -1 )         { // 終端記号までループ
                                        if ( pAns[ i + 0 ] == 0 ) {
                                                // 数値ならそのままスタックに入れる
                                                                StackPush( pAns[ i + 1 ] );
                                                } else {
                                                // 記号ならスタックからデータを 2 つ取り出し計算
                                                int iA,iB,iC;
                                                StackPop( &iB ); StackPop( &iA );
                                                switch( pAns[ i + 1 ] ) {
                                                    case REPAULAND_TYPE_ADD:
                                                                iC = iA + iB; break;
                                                    case REPAULAND_TYPE_SUB:
                                                                iC = iA - iB; break;
                                                    case REPAULAND_TYPE_MUL:
                                                                iC = iA * iB; break;
                                                    case REPAULAND_TYPE_DIV:
                                                                iC = iA / iB; break;
```

→ つぎのページに続く

```
                        case REPAULAND_TYPE_PAU:
                    iC = iA % iB; break;
                        case REPAULAND_TYPE_DOUBLE:
                                iC = (int)pow( iA, iB );
break;
                        case REPAULAND_TYPE_AND:
                                iC = iA & iB; break;
                        case REPAULAND_TYPE_OR:
                                iC = iA | iB; break;
                    case REPAULAND_TYPE_NOT:
                        // not の対象は 1 つだけなので，片方戻す

StackPush( iA );
                                if ( iB == 0 ) { iC = -1; }
                                else { iC = 0; }
                                break;
                        // ここから比較記号
                        case REPAULAND_TYPE_EQU_EQU:
                                if ( iA == iB ) { iC = -1; }
                            else { iC = 0; }
                                break;
                        case REPAULAND_TYPE_NOT_EQU:
                                if ( iA != iB ) { iC = -1; }
                                else { iC = 0; }
                                break;
                        case REPAULAND_TYPE_OVER_EQU:
                                if ( iA >= iB ) { iC = -1; }
                                else { iC = 0; }
                                break;
                        case REPAULAND_TYPE_MIN_EQU:
                                if ( iA <= iB ) { iC = -1; }
                            else { iC = 0; }
                            break;
                    case REPAULAND_TYPE_AND_AND:
                            if ( iA && iB ) { iC = -1; }
                            else { iC = 0; }
                            break;
                    case REPAULAND_TYPE_OR_OR:
                            if ( iA || iB ) { iC = -1; }
                            else { iC = 0; }
                            break;
                    case REPAULAND_TYPE_OVER:
                            if ( iA > iB ) { iC = -1; }
                    else { iC = 0; }
                            break;
                    case REPAULAND_TYPE_MIN:
                            if ( iA < iB ) { iC = -1; }
                            else { iC = 0; }
                            break;
                }           // of switch( c )
                // 計算結果をスタックに戻す
                StackPush( iC );
        }
        i += 2;
    }
    // スタックに残っている答えを取りだす
    StackPop( iAnswer );
  }  // of ここまでが逆ポーランドからの算出
  // 逆ポーランド算式データを解放する
  free( pAns );
  return dwCnt;
}

// 算式文字列を計算して答えを返す(読み進んだ文字数をreturn, 答えを nAnswerに返すように
変更 )
int GetIntFromString( char *pMath, int *nAnswer ) {
      return GetIntFromString_Sub( (unsigned char*)pMath, nAnswer );
}

/////////////////////////////////////////////////////////////////////////////
////////////////

// SHIFT-JIS漢字コードか判別
int IsKanji( unsigned char cData ) {
      if ( cData < 0x81 ) { return 0; }
      if ( cData < 0xa0 ) { return 1; }
      if ( cData < 0xe0 ) { return 0; }
      if ( cData < 0xff ) { return 1; }
      return 0;
}

// プログラムはここから始まる
void main() {
      FILE *fp; int nSize;
      unsigned char *p;
      int nPnt = 0;
      // 初期化
      ValueSysInit();
      // マクロファイルをメモリに読み込む
      fp = fopen( "macro.txt", "rb" );
```

```
      fseek( fp, 0, SEEK_END ); nSize = ftell( fp ); fseek( fp, 0, SEEK_SET );
      p = ( unsigned char* )malloc( nSize + 1 );
      fread( p, 1, nSize, fp );
      fclose( fp );
      // 読んだデータの終端に¥0をつける
      p[ nSize ] = '¥0';

      // interpreter /////////////////////////////////////////////////////////
/////
      while( nPnt < nSize ) {
                  if ( IsKanji( p[nPnt] ) ) {
                          // SHIFT-JIS漢字コードなので2byte進める
                          nPnt += 2;
                  } else {
                          if ( p[nPnt] == ':' ) {
                                  // コマンドマーク有
                                  nPnt++;
                                  if ( p[nPnt] == '$' ) {
                                          // 変数代入
                                          char szValueName[32];
                                          int nValue;
                                          nPnt++;
                                          nPnt +=
ValueSysGetValueName( (char*)&p[nPnt], szValueName, sizeof(szValueName) );
                                          while( p[nPnt] != '='
) { nPnt++; } nPnt++;
                                          nPnt +=
GetIntFromString( (char*)&p[nPnt], &nValue );
                                          ValueSysSetData(
szValueName, nValue );
                                  } else {
                                          // コマンド処理
                                          char szCmd[32]; int i
= 0;
                                          while( ( 'A' <=
p[nPnt] && p[nPnt] <= 'Z' )
                                                      || ( 'a'
<= p[nPnt] && p[nPnt] <= 'z' ) ) {
                                                      szCmd[i]
= p[nPnt]; nPnt++; i++;
                                          }
                                          szCmd[i] = '¥0';
                                          if ( strcmpi( szCmd,
"PRINT" ) == 0 ) {
                                                      int
nAns;
                                                      nPnt +=
GetIntFromString( (char*)&p[nPnt], &nAns );
                                                      printf(
"%d¥n", nAns );
                                          }
                                  }
                          } else {
                                  // コマンドでないので1byte進める
                                  nPnt++;
                          }
                  }
          }

          // 解放
          ValueSysRelease();
          free( p );
}
```

待て，次号！

今回のプログラムで，すでにBMPの核と呼べる部分はできあがってしまいました。次回以降を待たなくても，「PRINT」の例を見ながら自分の好きなように命令を逐次追加していけると思います。



とはいえ，やっぱり最低限，条件判定やジャンプ命令は欲しいところですよね。

次回は，今回の四則演算ルーチンを用いて条件判定を実装していきます。さぁ，野望に向けて一歩前進っ！

(無　料)
フリーソフト
HSP完全マスター
Hot Soup Processor

楽々3Dプログラミング！

第2回

監修：おにたま

「MX形式の3Dモデルを作る」

講師：悠黒喧史



はじめてHSPと出合ったときは面白さのあまりフリーソフトなのにカンパ(千円)。その後，好きが高じてHSPを書籍出版までこぎつけた。HSP Usherを自称する。

今回のポイント

MXモデルの表示と作成方法



MXモデルを表示してみよう

今回は，前回までの単純な立方体モデルではなく，MX形式データの3Dモデルを表示してみましょう。

MX形式データとは，HSP独自の3Dモデルのデータ形式です。MXモデルは，立方体や平面板のようにスクリプトだけで作り出すことはできません。3Dモデリングソフトなどを使って，データファイルを作成する必要があります(作成方法については後述)。

ここではとりあえず，HSP2.55に同梱されているサンプルデータ([sample¥hgimg]フォルダにある[rb_010.mx]と[rb_001.bmp])を使用してスクリプトを組んでみましょう(リスト1)。

《リスト1》MXモデルの表示

```
#include "hgimg.as"

        hgini                           ; HGIMGの初期化
;-----------------------------------------------------------------
        mxload "rb_010"                 ; MXモデルファイル読込み(システム変数statにモデルIDが入る)
;-----------------------------------------------------------------
        regobj ob,stat                  ; オブジェクト作成
        selpos ob :objset3 0,0,-50      ; ちょっと距離を離す
;-----------------------------------------------------------------
        gsel 0
        hgdraw                          ; 全オブジェクト描画処理
        hgsync 10                       ; ウェイト
        stop
```

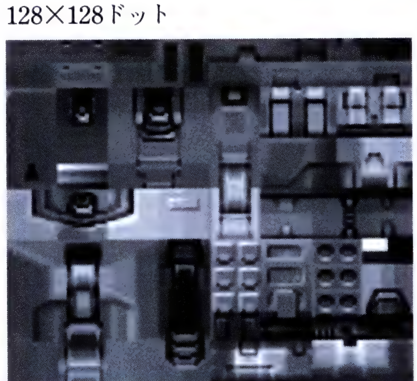
MXモデルデータ

rb_010.mx

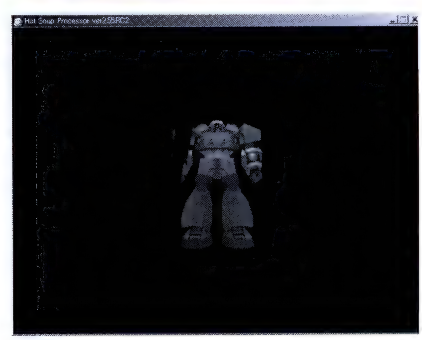
MXモデルに貼るテクスチャ

rb_001.bmp

128×128ドット



実行画面



・リスト1の解説

モデルの登録は[mxload]命令一つで，スクリプト自体は立方体のときよりもずっとシンプルになりました。これはモデルのデザインやサイズ，貼り付けるテクスチャの指定など，モデルの情報が全てMXデータに含まれているためです。

使用した命令

mxload " " (" " 内の MXファイルを読み込み，モデル登録する)

MXモデルを動かしてみよう

さて，前回同様，このMXモデルを動かしてみることにしましょう。モデル登録の部分以外は，立方体の場合と全く同じ処理です。

MXモデルを動かす(リスト2)

実行画面



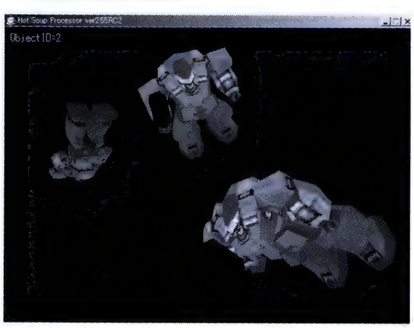
カーソルの[↑][↓][←][→]で上下左右，[Ctrl]を押しながらの[↑][↓]で前後に移動します。

テンキーの[4][6]でY軸回転，[8][2]でX軸回転，[1][0]でZ軸回転をします。

[Esc]でプログラムを終了します。

複数のMXモデルを動かす(リスト2s)

実行画面



カーソルの[↑][↓][←][→]で上下左右，[Ctrl]を押しながらの[↑][↓]で前後に移動します。

テンキーの[4][6]でY軸回転，[8][2]でX軸回転，[1][0]でZ軸回転をします。

《リスト2》MXモデルを動かす

```
#include "hgimg.as"

        hgini                                   ; HGIMGの初期化
;-----------------------------------------------------------------
        mxload "rb_010"                         ; MXﾓﾃﾞﾙﾌｧｲﾙ読込み
;-----------------------------------------------------------------
        regobj ob,stat                          ; ｵﾌﾞｼﾞｪｸﾄ作成
        selpos ob :objset3 0,0,-50              ; ちょっと距離を離す
;-----------------------------------------------------------------
        gsel 0
*main
        selpos ob                               ; MOC座標にｱｸｾｽ

        stick ky,79                             ; ｷｰｽｷｬﾝ

        if ky&1 :objaddf1 0,-0.1f               ; X軸移動
        if ky&4 :objaddf1 0,0.1f                ;
        if ky=2 :objaddf1 1,-0.1f               ; Y軸移動
        if ky=8 :objaddf1 1,0.1f                ;
        if ky=66 :objaddf1 2,-0.1f              ; Z軸移動
        if ky=72 :objaddf1 2,0.1f               ;

        selang ob                               ; MOC角度にｱｸｾｽ

        k=-1                                    ; ﾃﾝｷｰ入力ﾁｪｯｸ
        repeat 10                               ;
                getkey a,96+cnt                 ;
                if a=1 :k=cnt :break            ;
        loop                                    ;

        if k=8 :objaddf1 0,-0.1f                ; X軸回転
        if k=2 :objaddf1 0,0.1f                 ;
        if k=4 :objaddf1 1,-0.1f                ; Y軸回転
        if k=6 :objaddf1 1,0.1f                 ;
        if k=1 :objaddf1 2,-0.1f                ; Z軸回転
        if k=0 :objaddf1 2,0.1f                 ;

        hgdraw                                  ; 全ｵﾌﾞｼﾞｪｸﾄ描画処理
        hgsync 10                               ; ｳｪｲﾄ

        if ky!128 :goto *main                   ; [Esc]が押されるまでﾙｰﾌﾟ

        hgbye                                   ; HGIMGを解放
```

《リスト2.S》複数のMXリストを動かす

```
#include "hgimg.as"

        hgini                                   ; HGIMGの初期化
;-----------------------------------------------------------------
        mxload "rb_010"                         ; MXﾓﾃﾞﾙﾌｧｲﾙ読込み
        md=stat                                 ; 変数mdにﾓﾃﾞﾙIDを代入
;-----------------------------------------------------------------
        repeat 3
                regobj ob.cnt,md                ; ｵﾌﾞｼﾞｪｸﾄ作成
                selpos ob.cnt :objset3 cnt-1*20,0,-50  ; ちょっと距離を離す
        loop
;-----------------------------------------------------------------
        gsel 0 :i=1
*main
        selpos ob.i                             ; MOC座標にｱｸｾｽ

        stick ky,79 :if ky=1024 :i=i+1¥3        ; ｵﾌﾞｼﾞｪｸﾄIDの切換え

        if ky&1 :objaddf1 0,-0.1f               ; X軸移動
        if ky&4 :objaddf1 0,0.1f                ;
        if ky=2 :objaddf1 1,-0.1f               ; Y軸移動
        if ky=8 :objaddf1 1,0.1f                ;
        if ky=66 :objaddf1 2,-0.1f              ; Z軸移動
        if ky=72 :objaddf1 2,0.1f               ;

        selang ob.i                             ; MOC角度にｱｸｾｽ

        k=-1                                    ; ﾃﾝｷｰ入力ﾁｪｯｸ
        repeat 10                               ;
                getkey a,96+cnt                 ;
                if a=1 :k=cnt :break            ;
        loop                                    ;

        if k=8 :objaddf1 0,-0.1f                ; X軸回転
        if k=2 :objaddf1 0,0.1f                 ;
        if k=4 :objaddf1 1,-0.1f                ; Y軸回転
        if k=6 :objaddf1 1,0.1f                 ;
        if k=1 :objaddf1 2,-0.1f                ; Z軸回転
        if k=0 :objaddf1 2,0.1f                 ;

        hgdraw                                  ; 全ｵﾌﾞｼﾞｪｸﾄ描画処理

        color 255,255,255                       ; ｵﾌﾞｼﾞｪｸﾄIDの表示
        pos 8,8 :mes "ObjectID="+ob.i           ;

        hgsync 10                               ; ｳｪｲﾄ

        if ky!128 :goto *main                   ; [Esc]が押されるまでﾙｰﾌﾟ

        hgbye                                   ; HGIMGを解放
```

Tab で対象となるオブジェクトIDを切り換えます。

Escでプログラムを終了します。

・リスト2の解説

モデルの種類が変わっただけで，プログラムそのものは前回のおさらいに近いものです。オブジェクトコントロールは3Dプログラミングの中核となるものですから，しっかりと理解しておいてください。

オブジェクトの向きに進む移動処理

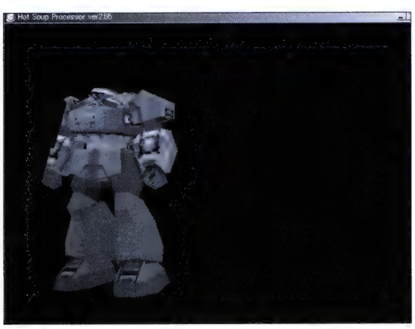
ここまで，MOC座標に加減算してオブジェクトを移動，MOC角度に加減算してオブジェクトを回転させる方法を解説してきました。

これでもう自由自在にオブジェクトをコントロールできそうなのですが，もうひとつ，オブジェクトを動かすプログラムテクニックを伝授しておきましょう。

次の方法では，そのオブジェクトの向いている自由な方向へ，簡単に移動させることができます。ラジコンカーを動かしているような感覚の移動です。

向いてる方向に進む(リスト3)

実行画面



カーソルの← → で左右に回転，↑ ↓ で前進・後退します。

Escでプログラムを終了します。

・リスト3の解説

[objgetfv]命令は，現在，対象となっているMOC情報の値を変数に代入します。

スクリプト記述例

selang ob :objgetfv an

この例では，変数obの示すオブジェク

《リスト3》向いている方向に進む

```
#include "hgimg.as"
            hgini                               ; HGIMGの初期化
;-----------------------------------------------------------------
            mxload "rb_010"                     ; MXﾓﾃﾞﾙﾌｧｲﾙ読込み
;-----------------------------------------------------------------
            regobj ob,stat                      ; ｵﾌﾞｼﾞｪｸﾄ作成
            selpos ob :objset3 0,0,-50      ; ちょっと距離を離す
;-----------------------------------------------------------------
            gsel 0
*main
            stick ky,15                         ; ｷｰｽｷｬﾝ

            selpos ob :objgetfv po              ; MOC座標をpo.0～2に得る

            selang ob :objgetfv an              ; MOC角度をan.0～2に得る

            if ky&1 : fsub an.1,0.05f           ; 左回転
            if ky&4 : fadd an.1,0.05f           ; 右回転

            objsetfv an                         ; MOC角度をｾｯﾄ

            if ky&2 { fvdir an,0.0f,0.0f,1.0f   ; 前進
                        fvadd po,an.0,an.1,an.2 ;
            }
            if ky&8 { fvdir an,0.0f,0.0f,-1.0f  ; 後退
                        fvadd po,an.0,an.1,an.2 ;
            }
            selpos ob :objsetfv po              ; MOC座標をｾｯﾄ

            hgdraw :hgsync 10                   ; 描画 / ｳｪｲﾄ

            if ky!128 :goto *main               ; [Esc]を押すまでﾙｰﾌﾟ

            hgbye
```

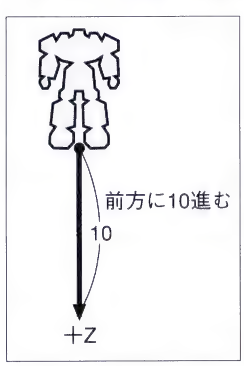
トのMOC角度のX,Y,Zの値を変数an.0,an.1,an.2に代入します。値を代入した変数an.0～an.2は，通常の配列変数と同様に扱うことができます。

このオブジェクトの移動処理では，[fvdir]命令の使い方がポイントとなっています。

[fvdir]命令は，与えられた角度値と移動量から新たな座標値を計算する命令です。つまり，指定した角度で，指定した分だけ移動した先の座標値を算出してくれます。

オブジェクトのMOC座標を使い，動かしたい分の移動量を指定すれば，移動させる先の座標値を知ることができます。この座標値を移動量として，現在のオブジェクトに加えることで，オブジェクトを自身の向きに対応して移動させることができます。

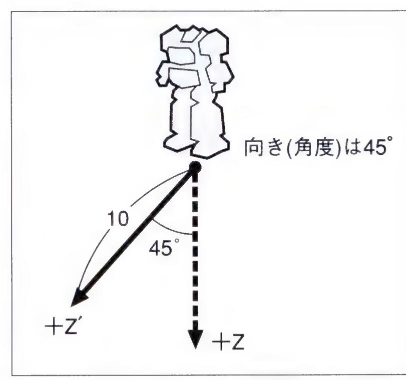
例えば，Y軸に45度(ラジアンで0.78)傾いたオブジェクトを前方(オブジェクト自身の+Z方向)に10進ませる場合，



移動量

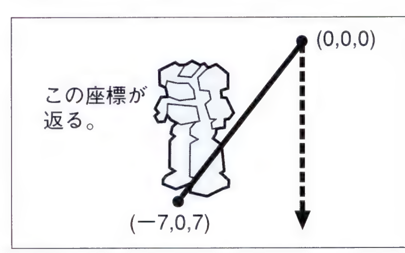
X移動量=0
Y移動量=0
Z移動量=10

MOC角度



X軸角度=0
Y軸角度=0.78
Z軸角度=0

移動先の座標(計算結果)



X座標=-7
Y座標=0
Z座標=7

変数an.0～an.2にオブジェクトのMOC角度が入っているとすれば，次の

ようになります。

スクリプト記述例

fvdir an,0.0f,0.0f,10.0f

ちょっと混乱しやすいのが，[fvdir]命令の計算結果が，直前までMOC角度値の入っていた変数an.0～an.2に返るという点です。変数anの内容は，MOC角度値ではなくなってMOC座標値が入ってくるので，それでMOC座標に移動成分として加算できるわけです。[fvadd]命令で，この変数an.0～an.2をMOC座標値の変数po.0～po.2に加算しています。

最後に[objsetfv]命令で，この変数po.0～po.2をオブジェクトのMOC座標にセットすれば，オブジェクトの向きに対応した移動処理が完了します。

少し概念が難しいかもしれませんが，プログラムに慣れてくると3Dゲームでは，とても重宝する移動処理テクニックです。よくわからないままでも，使っているうちに理解できてくるでしょう。

使用した命令

objgetfv…… 変数(配列0,1,2)にMOC情報X,Y,Zを代入する
objsetfv…… MOC情報X,Y,Zに変数(配列0,1,2)の内容をセット
fvdir………… 角度と移動量から新たな座標を割り出す
fvadd………変数(配列0,1,2)に三つの小数値を一度に加算
fadd………… 小数値の加算
fsub………… 小数値の減算

MX形式のモデルを作成できるツール

① YJ Studio 2001 モデリング編集ツール

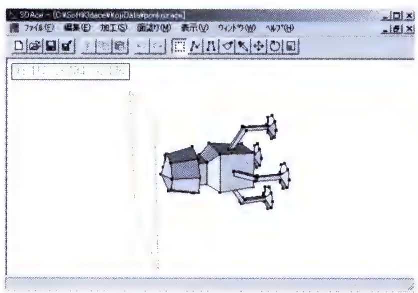
モデリング編集およびテクスチャの設定ができます。

[YJPコンバータ]によるYJP形式を介して，LWO,X,ROK,DXF形式のデータをMX形式にもっていくことも可能です。[YJPコンバータ]はシェアウェアですが，YJP形式を使うぶんには機能制限はありません。

よし～ふ ゲーム開発部屋

http://member.nifty.ne.jp/Yoshirf/

② 3DACE

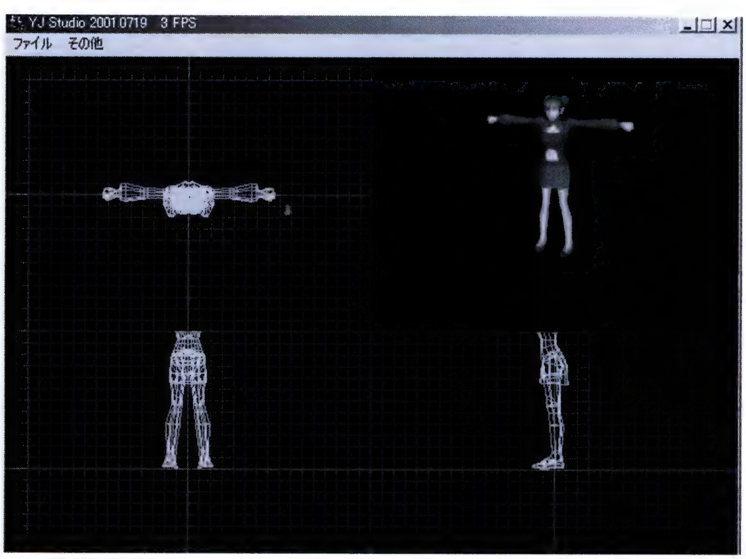


フリーのモデリングソフトはいくつかありますが，ビギナーでも直感的に扱えるソフトです。

[ace2mx]というコンバータでHSPのMX形式データを作成することができます。

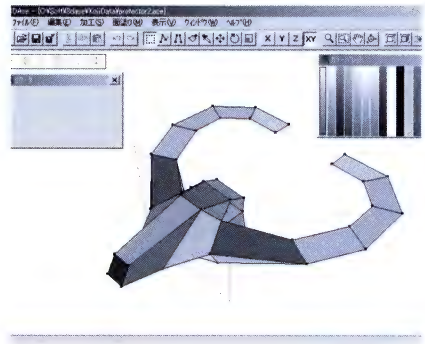
3DACEの部屋

http://hp.vector.co.jp/authors/VA017881/



3DACEでお手軽にHSP用のMXモデルを作ってみよう！

まず，普通に3DACEでモデルを作成し，別々のテクスチャを張りたい面を



色分けして塗ります。

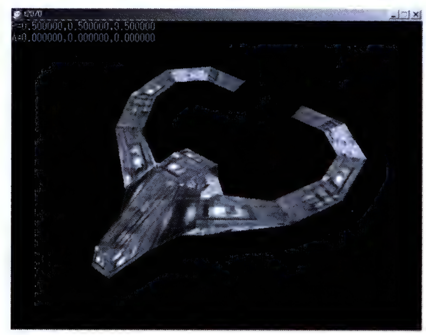
保存したACE形式のデータを[ace2mx]でMX形式にコンバートします。

すると，MXファイル名と同名のBMPファイルが作成されます。



このBMP画像を描き変えることで，テクスチャを変更することができます。





HGIMG-FAQ

質問 

[HGIMG]の命令を一つ一つ解説しているマニュアルはありませんか？

答え 

ヘルプの命令索引や命令リファレンスにも説明がありますが，HSPの[docs]フォルダにある[hgimg.txt]というテキストファイルにHGIMGの各命令が詳しく解説されています。

プリントアウトして手元に置いておくと便利ですよ。

☆　☆　☆

■ゲーム・フィールドを自動的に迷路にするテクニック！



プログラマーをめざす人の必見講座

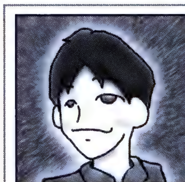
図解! Visual Basic ゲーム作成術

最終回　キャラクター・アクション・ゲーム番外編　迷路の作成

キャラクター・アクション・ゲームでは，さまざまなキャラクターが活躍する舞台として，ゲーム・フィールドが表示されますね。このゲーム・フィールドを迷路状にして，迷路を解く要素をゲームに付け加える場合があります。このような場合，迷路を何種類か作りださなければなりません。

今回は，迷路を自動的に生成するためのアルゴリズムを紹介し，『Microsoft（マイクロソフト） Visual Basic（ビジュアル ベーシック）（以下VBと略）』でサンプルを作成してみましょう。

TEXT&PROGRAMMING●東　幸太



BASIC歴15年以上のプログラマー。環境配慮の時代，パソコンもリサイクル対象となるようだ。再利用できる部品をしっかりと活用することも大切だが，同時にパソコン自体を長く大切に使うことも重要だ。古いパソコンも，メモリを256Mバイトにして高速ハードディスクを付ければ，かなりの速度アップができる。捨てる前に，パワーアップを考えよう！

Microsoft Visual Basic Programming System for Windows Ver.6.0(日本語版)

オープン価格

※編集部の価格調査によると，下記価格が目安になります。

ラーニング版2万円強

(アカデミック・パッケージ1万円弱)

プロフェッショナル版8～9万円

(アカデミック・パッケージ1万円強)

問い合わせ先

マイクロソフト

☎03-5454-2300

ホームページ

http://www.microsoft.com/japan/

動作環境

Windows 95／98／2000／NT4.0が動作するマシン



ゲームと迷路は深い関係がある！

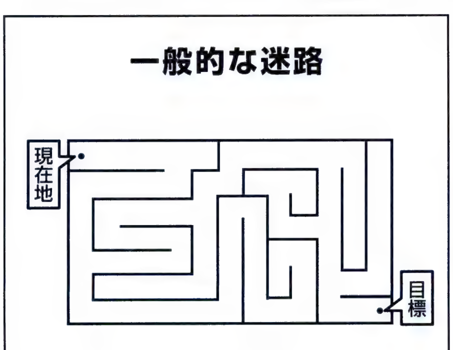
まずは，第1図を見てください。このようにパッと見ただけでは目的地までのルートがはっきりしない迷路は，それ単体でもゲーム的な要素を持っています。

実際のゲームをこの迷路と組み合わせることで，ゲームに複雑さを追加するとともに，先の読みにくいゲームにできます。とくに，目的地までのルートが1種類しかない迷路の場合は，正解ルートを見つけるのがむずかしいため先が読めず，必然的に試行錯誤が増えていきますね。この増加する試行錯誤をうまくゲームに取り込むことで，ゲームの難易度を調整できるわけです。

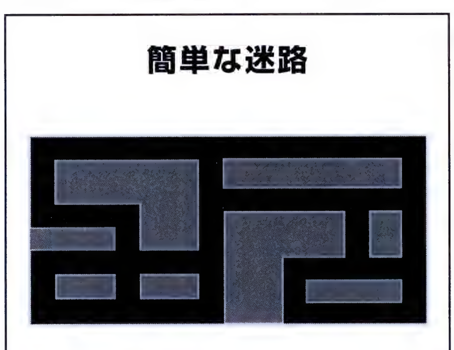
また，ここまでむずかしい迷路ではなく，ゲーム・フィールドを軽く迷路状にする場合もあります（第2図）。毎回同じステージにならないように乱数で迷路を作りだせば，ゲームのパターン化を防げます。ゲームがパターン化するのを防ぐことで，パターンを覚えれば楽にクリアできるというゲームではなく，そのゲームに慣れることでクリアできるようになるゲームに仕上げられます。

複雑な迷路による試行錯誤や，パターン化されないゲーム・フィールドは，どちらも使いかたをまちがえるとゲーム・デザインに大きな悪影響をあたえます。あまりにも退屈な試行錯誤が多くなるようでは問題がありますし，あまりにもランダムなゲーム・フィールドはゲームに慣れることがむずかしくなってしまい，プレイヤーの技術でなくゲーム・フィールドのできぐあいでゲームがきまってしまうことになります。

このように，迷路は正しく使わないと，単調なゲームになったり習熟するのに時間がかかるゲームになったりしてしまいます。しかし，うまく使えば，先の読めない未知の世界を演出できるのです（第3図）。



《第1図》「目標まで何秒でたどり着けるか」など，迷路にはそれ自体にゲーム性がある



《第2図》通常のキャラクター・アクション・ゲームは，この程度の迷路が多いだろう

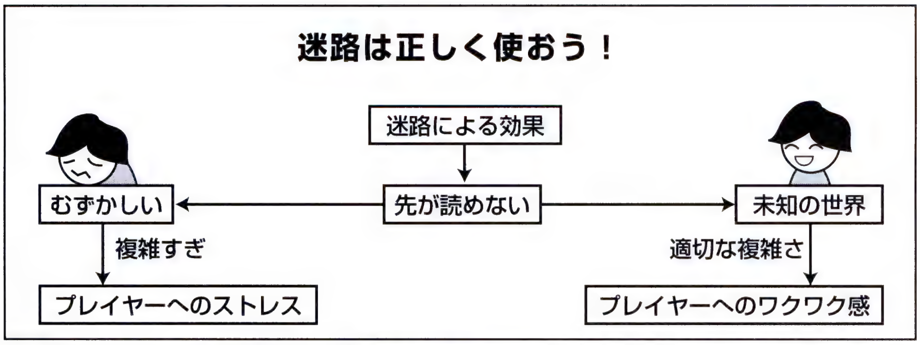
壁作成方式と穴掘り方式

迷路を作成する方式には，主につぎの2種類があります。

①壁を作っていく方法（壁作成方式）

②通路を作っていく方法（穴掘り方式）

どちらを選択するかは，簡単になるほうを使えばよいのですが，ゲーム・フィールドの構成によってどちらが簡単にな



《第3図》たとえば「時間制限をなくして複雑な迷路を楽しんでもらう」「敵キャラクターをワナにかける路地を活用してもらう」というように，うまく迷路を使いわけよう

るかは変わっていきます。

①の壁作成方式は，最初は通路ばかりのゲーム・フィールド上で，基準点の周囲に壁を作りながら基準点を移動させていくというものです（第4図）。壁が薄く，仮想画面上に存在しないようなゲームの場合に適しています。

②の穴掘り方式は，最初は壁ばかりのゲーム・フィールド上で，基準点に通路を作りながら基準点を移動させていくというものです（第5図）。壁が，仮想画面上で一つのキャラクターとして存在するタイプのゲームに適しています。

壁作成方式は壁が仮想画面上に存在しないため，ゲーム・フィールド上の迷路の密度を高くできます。しかし，仮想画面によって壁の存在に関する情報も処理する必要がでてくるため，ややプログラムが複雑になります。

穴掘り方式は，壁を一つのキャラクターとして扱うために，ゲーム・フィールド上の何割かは壁で埋まってしまいます。そのため，迷路の密度は低くなりますが，プログラムは比較的簡単になります。

今回は，プログラムが比較的簡単になる，②の穴掘り方式について説明しましょう。

穴掘り方式は具体的にどう処理する？

ゲーム・フィールドに迷路を作成するというのは，プログラム的には壁を示す仮想画面に迷路を作成することになります。第6図のような仮想画面の中で，迷路を作成することを考えてみましょう。

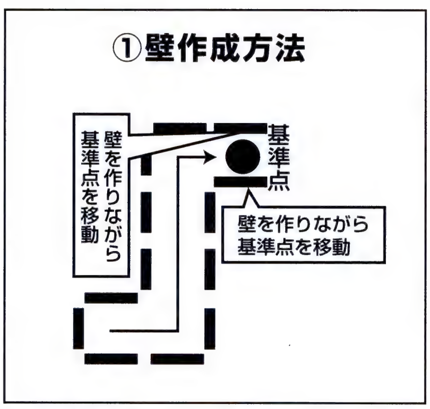
まず，仮想画面をすべて壁で埋めつくし，そのあとでスタート地点に穴を掘っていく空白を置きます。この空白を移動させることで，壁の中に通路を掘っていくわけです。

移動方向は乱数で決定し，一度に2キャラクターぶん移動させることにします（第6図）。移動した結果，すでに掘ってある場所や，ゲーム・フィールド外部に来てしまう場合は移動できません。移動先がゲーム・フィールド内部の壁になっている場合だけ，移動できるのです。

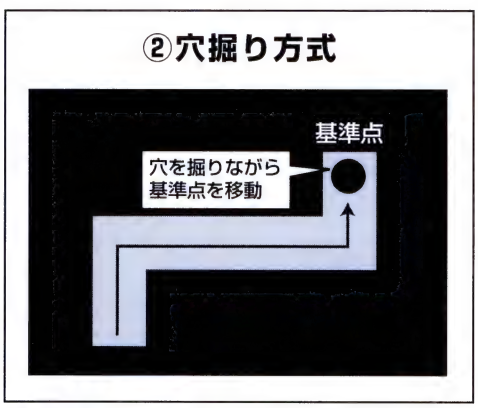
移動できる方向が複数存在する場合は，その座標を起点データ配列変数に記憶しておきます（第7図）。移動できる方向がない場合は，それ以上掘り進むのをやめ，起点データ配列変数に残っている場所から新たに掘り進めます。

最終的に，起点データ配列変数のデータがなくなれば，それ以上掘れる場所がないことになり，迷路ができあがったことになります。

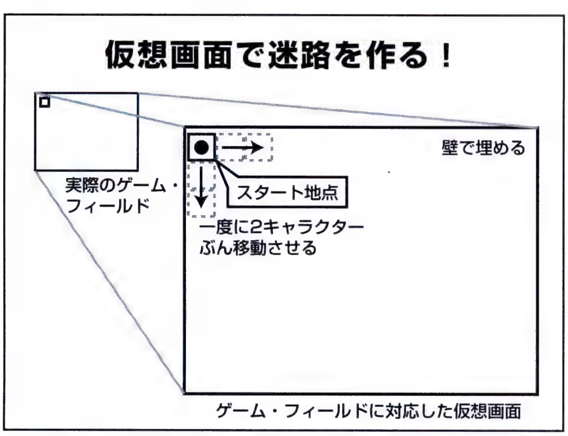
このようにしてできた迷路は，スタートからゴールまでのルートが1とおりしか存在しない迷路になります。これに対し，一部の壁を空白に置き換えてやると，第8図のように部分的にループ（繰り返し）を含む迷路になります。



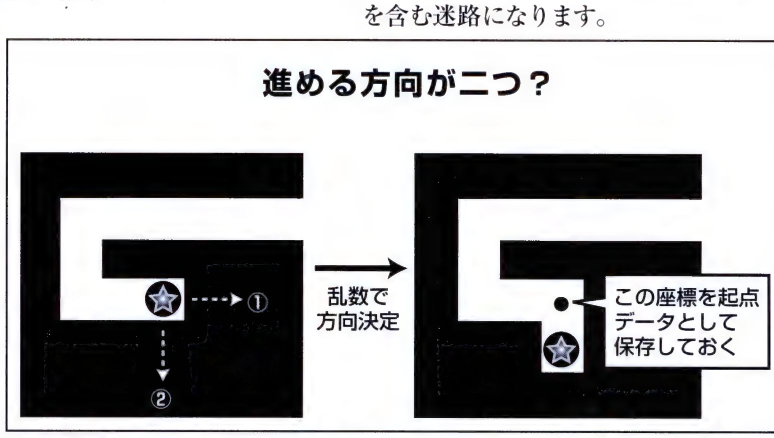
《第4図》壁を作りながら基準点をどんどん移動させて，迷路を完成させていく



《第5図》壁だけのゲーム・フィールドを作って，穴を掘るように通路部分を作成していく

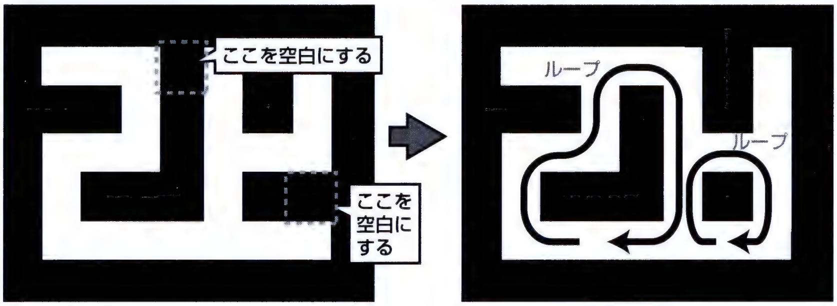


《第6図》今回は仮想画面を使って，穴掘り方式で迷路を作成する



《第7図》基準点の進める方向が複数ある場合は，起点データとして登録しておくこの起点データのストックがなくなれば，迷路は完成だ

プレイヤーの自由度をあげよう！



《第8図》穴掘り方式でできた迷路は，スタートからゴールまで一つの道筋となる。プレイヤーがきゅうくつに感じるようなら，このように空白を入れてループできるようにしよう

単純な迷路が必要な場合は，スタートからゴールまでが一本道のものをそのまま適用すればよいでしょう。しかし，一般的なゲーム・フィールドとして利用する場合は，一部の壁を取り除いて，ある程度ループを含ませることも必要となります。

VBで利用してみよう！

それでは，実際にVBでプログラミングしてみましょう。前号で紹介したキャラクター・アクション・ゲームに，今回の迷路アルゴリズムを組み込んでいます。**リスト1**が，そのゲーム『キャッチ・ドット・ゲームMaze（メイズ）』です。

リストを入力して実行すると，**第9図**のような画面が表示されます。基本ルールは，前号までのゲームと同じです。画面には，プレイヤーが操作するプレイヤー・キャラクターと，敵キャラクターが四つ登場します。敵キャラクターは，ランダムに格子状の通路を移動します。移動した後に，ドットを落としていく場合があります。

プレイヤーは，プレイヤー・キャラクターをテンキーの[2][4][6][8]で上下左右に移動させます。敵キャラクターが落としたドットを，制限時間内になるべくたくさん拾い集めてください。

ドットを取ると，その段階での残り時間に応じた得点が加算されます。残り時間が多いほど高得点になるわけです。ただし，敵キャラクターと接触すると，残り時間が大幅に減ってしまいます。接触された敵キャラクターは，しばらくのあいだ停止します。

ゲーム・フィールド内には，プレイヤー・キャラクターを一定時間高速化するスピード・アップ・キャラクターが移動しています。プレイヤー・キャラクターがこのキャラクターに触れると，それ以後一定時間，移動速度が高速化されます。

また，敵キャラクターは，まれに大きなドット（ビッグドット）を落としていきます。このドットを取ると反撃モードになり，プレイヤー・キャラクターと敵キャラクターの関係が逆転します。反撃モードのあいだに敵キャラクターと接触すると，敵キャラクターが消滅し，得点が加算されます。一定時間が経過すると，反撃モードは終了します。

ベーマガ・ホームページ ダウンロードの方法

まずは，以下のアドレスにインターネット・アクセスします。

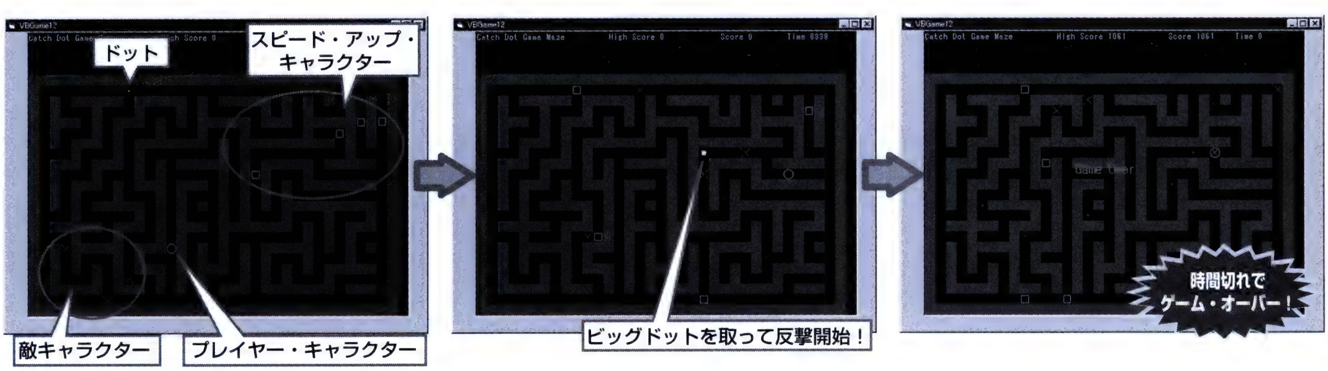
●http://www.basicmagazine.net/

そこから，「連載プログラミング講座」をクリックしてください。

そして，『図解！ VBゲーム作成術』の画面の下にある「ファイルをダウンロード」をクリックします。すると，「ファイルのダウンロード」ウィンドウが表示されるので，「このファイルをディスクに保存する」をチェックして[OK]ボタンを押します。「名前を付けて保存」ウィンドウが表示されるので，適当な場所（c:¥bm03など）に保存します。同じ場所に“lha.exe”をコピーして，[スタート]-[プログラム]-[MS-DOSプロンプト]で，その場所に移動します（上記の例だと「cd c:¥bm03」と入力）。そして，「lha e z2002-03」とキーボードから入力して，[Enter]キーを押します。これで，解凍作業は終わりました（Windowsのツールで解凍してもOK）。

“readme.txt”には，ファイルの概要が書かれていますので，参照してください。

これが『キャッチ・ドット・ゲームMaze』だ！



《第9図》『キャッチ・ドット・ゲームMaze』。基本的にはこれまでと同じゲームであるが，ゲーム・フィールドが迷路になったため，プレイヤー・キャラクターをスムーズに移動させることがむずかしくなっている。二つのキー入力を受け付けるので，カーブの向きをあらかじめ入力する先行入力を駆使しよう

VBでの迷路アルゴリズムに注目！

リスト1では，迷路状のゲーム・フィールドを作成するために，これまで説明した穴掘り方式を利用しています。そのままでは一本道の迷路になってしまうため，ゲーム・フィールドの外周部分は強制的に通路とし，通路のループができるようにしています。

迷路作成部分のフローチャートは，第10図のようになっています。

基準となる座標は，x0とy0に格納されています。この座標を基準とし，chkmazemove関数を使って，掘り進める方向を配列変数hlにリストアップしていきます。候補の中から，実際に掘り進む方向を乱数で決定して掘り進み，他に候補が残っている場合は配列変数spxとspyに起点データとして座標を保存します。

掘り進める方向がない場合は，spccカウンタをもとに起点データを参照して基準座標x0とy0を更新し，新しい場所から掘り進みます。掘り進める方向がなく，spccカウンタが0になっていれば，すべての掘り進める方向をチェックしたことになるので終了します。

chkmazemove関数は，指定された場所から指定された方向に掘り進めるかどうかを判断する関数です。最初に，座標をチェックしてゲーム・フィールドからはみださないことを確認してから，座標を進めた場所が壁かどうかを調べています。移動方向の数値化，移動量のテーブル化

(dx(), dy())

などにより，簡単にまとまっています。

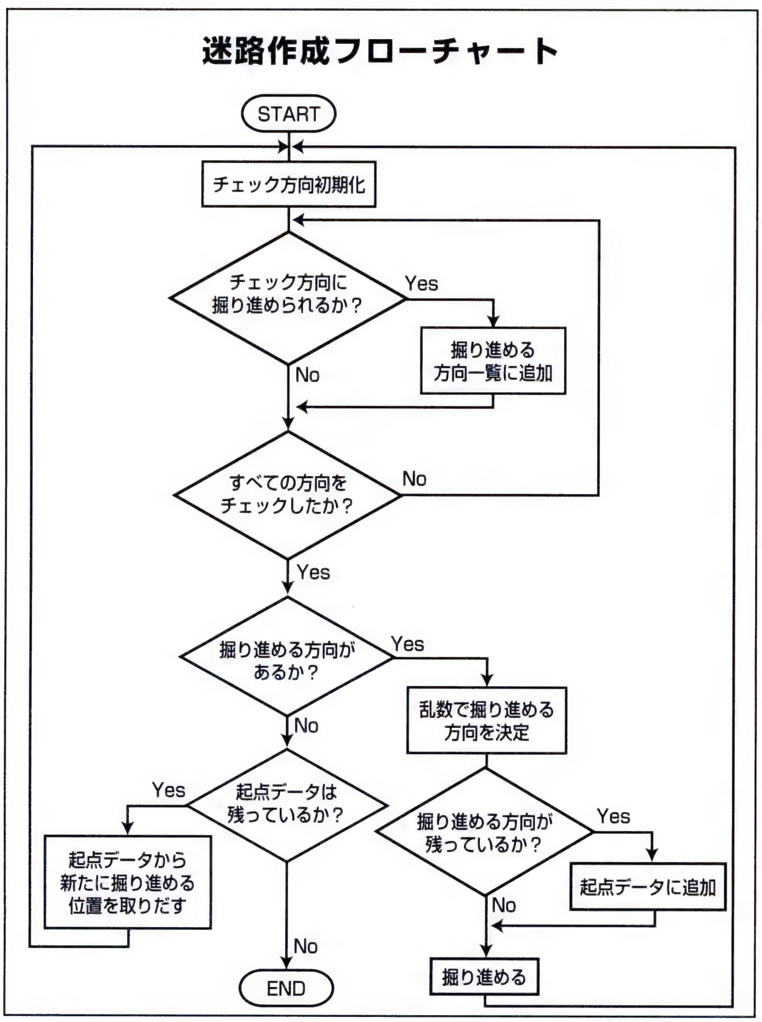
ところで，ゲーム・フィールドが複雑になったため，敵キャラクターなどの移動処理も変更する必要はあるでしょうか？　答は「いいえ」です。これまでに作ってきた追跡アルゴリズムなどは，ゲーム・フィールドの形状に影響されないものになっているため，ほぼそのまま利用できます。

おわりに

単純で見とおしの悪い迷路というのは，基本的にプレイヤーにストレスをあたえる要素を持っています。このため，そのままゲームに応用するのはむずかしい面があります。

しかし，壁を多めに取りのぞいてループを増やしてやれば，変化に富んだゲーム・フィールドになります。ゲーム・デザインに応じて，程よい迷路を作ってみてください。

来月からは，装いも新たに新連載がはじまります。お楽しみに！



《第10図》リスト1では，このフローチャート（流れ図）のように迷路を作成している

《第1表》『キャッチ・ドットゲームMaze』の主な変数

変数名	内容
hiscore	ハイスコア
score	プレイヤー・スコア
px	プレイヤー・キャラクターX座標
py	プレイヤー・キャラクターY座標
pm	プレイヤー・キャラクター移動カウンタ初期値
pmc	プレイヤー・キャラクター移動カウンタ
spupc	スピード・アップ期間カウンタ
modec	パワーアップ期間カウンタ
ef ()	敵キャラクター状態フラグ
ex ()	敵キャラクターX座標
ey ()	敵キャラクターY座標
eh ()	敵キャラクターの方向
em ()	敵キャラクター移動カウンタ初期値
emc ()	敵キャラクター移動カウンタ
sf ()	スピード・アップ・キャラクター状態フラグ
sx ()	スピード・アップ・キャラクターX座標
sy ()	スピード・アップ・キャラクターY座標
sh ()	スピード・アップ・キャラクターの方向
sm ()	スピード・アップ・キャラクター移動カウンタ初期値
smc ()	スピード・アップ・キャラクター移動カウンタ
dx ()	方向→X座標移動量変換テーブル
dy ()	方向→Y座標移動量変換テーブル
eht ()	敵キャラクター＆スピード・アップ・キャラクター移動方向検索順テーブル
dp ()	キャラクター位置関係テーブル
kg1 ()	敵キャラクター用仮想画面（0：存在しない，それ以外：存在するキャラクターの添え字+1）
kg2 ()	スピード・アップ・キャラクター用仮想画面（0：存在しない，それ以外：存在するキャラクターの添え字+1）
kg3 ()	静止キャラクター用仮想画面（0：空白，1：壁，2：ドット，3：ビッグドット）
kg4 ()	敵キャラクター追跡用仮想画面
spx ()	迷路作成用起点データX座標
spy ()	迷路作成用起点データY座標
spcc	起点データ・カウンタ
ti	残り時間
dotc	ドットを発生させる間隔計測カウンタ
gamestat	ゲーム進行状況（0：ゲーム中，1：ゲーム・オーバー）

《リスト1》『キャッチ・ドットゲームMaze』のプログラム・リスト（VB6.0用）

```
Option Explicit

Dim keystat%(9) 'テンキーの状態を記録する変数
Dim keytrig0%, keytrig1%, keytrig2%, keytrig3% 'トリガー用キーの状態を
記録する変数
Dim px%, py%, ph%, pf%, pmc%, pm%, pm2%, spupc% 'プレイヤーキャラクタ
Dim ex%(3), ey%(3), eh%(3), ef%(3), emc%(3), em%(3) '敵キャラクタ
Dim sx%(3), sy%(3), sh%(3), sf%(3), smc%(3), sm%(3) 'スピードアップキ
ャラクタ
Dim kg1(35, 26) '仮想画面１（敵キャラクタ用）
Dim kg2(35, 26) '仮想画面２（スピードアップキャラクタ用）
Dim kg3(35, 26) '仮想画面３（ドット＆壁用）
Dim kg4(3, 35, 26) '仮想画面４（敵キャラクタ追跡用）
Dim eht(3, 3, 2) '敵キャラクタ移動方向検索テーブル
Dim dx(3), dy(3) '方向→移動量変換テーブル
Dim dp(3, 2, 2) 'キャラクタ位置関係テーブル
Dim hiscore%, score% 'ハイスコアとスコア
Dim ti% '残り時間
Dim dotc% 'ドットを発生させる間隔を数えるカウンタ
Dim modec% 'パワーアップ期間を数えるカウンタ
Dim gamestat% 'ゲーム進行状況

Private Sub Form_Load()
    hiscore = 0
    gameinit
End Sub

Private Sub gameinit() 'ゲーム初期化
    Dim ox%, oy%, x%, y%, xx%, yy%, i%, j%, f%, h%, c0%, c1%
    Dim mef%, x0%, y0%, spcc%, t%, p%
    Dim hl%(4), spx%(500), spy%(500)

    With Form1
        .Width = Screen.TwipsPerPixelX * 640 'フォームのサイズを設定
        .Height = Screen.TwipsPerPixelY * 480
        .Caption = "VBGame12"
    End With
    With Picture1
        .Width = Screen.TwipsPerPixelX * 576 'ピクチャボックスのサイズ
を設定
        .Left = Screen.TwipsPerPixelY * 32
        .Height = Screen.TwipsPerPixelY * 432
        .Top = 0
        .ScaleMode = 3
        .ScaleHeight = 432
        .ScaleWidth = 576
        .BackColor = RGB(0, 0, 0)
        .ForeColor = RGB(255, 255, 255)
        .FillStyle = 0
        .Font = "ＭＳ ゴシック"
        .FontSize = 10
    End With
    Picture1.Cls
    Form1.Show

    dx(0) = 0: dy(0) = -1 '移動方向→移動量のテーブル
    dx(1) = 1: dy(1) = 0
    dx(2) = 0: dy(2) = 1
    dx(3) = -1: dy(3) = 0

    dp(0, 0, 0) = 3: dp(0, 0, 1) = 3: dp(0, 0, 2) = 3 '位置関係テーブ
ル
    dp(0, 1, 0) = 0: dp(0, 1, 1) = 0: dp(0, 1, 2) = 2
    dp(0, 2, 0) = 1: dp(0, 2, 1) = 1: dp(0, 2, 2) = 1

    dp(1, 0, 0) = 3: dp(1, 0, 1) = 0: dp(1, 0, 2) = 1
    dp(1, 1, 0) = 3: dp(1, 1, 1) = 0: dp(1, 1, 2) = 1
    dp(1, 2, 0) = 3: dp(1, 2, 1) = 2: dp(1, 2, 2) = 1

    dp(2, 0, 0) = 1: dp(2, 0, 1) = 1: dp(2, 0, 2) = 1
    dp(2, 1, 0) = 2: dp(2, 1, 1) = 0: dp(2, 1, 2) = 0
    dp(2, 2, 0) = 3: dp(2, 2, 1) = 3: dp(2, 2, 2) = 3

    dp(3, 0, 0) = 1: dp(3, 0, 1) = 0: dp(3, 0, 2) = 3
    dp(3, 1, 0) = 1: dp(3, 1, 1) = 0: dp(3, 1, 2) = 3
    dp(3, 2, 0) = 1: dp(3, 2, 1) = 2: dp(3, 2, 2) = 3

    '敵キャラクタ移動方向検索順テーブル
    eht(0, 0, 0) = 0: eht(0, 1, 0) = 1: eht(0, 2, 0) = 3: eht(0, 3, 0)
= 2
    eht(1, 0, 0) = 1: eht(1, 1, 0) = 0: eht(1, 2, 0) = 3: eht(1, 3, 0)
= 2
    eht(2, 0, 0) = 0: eht(2, 1, 0) = 3: eht(2, 2, 0) = 1: eht(2, 3, 0)
= 2
    eht(3, 0, 0) = 3: eht(3, 1, 0) = 0: eht(3, 2, 0) = 1: eht(3, 3, 0)
= 2
    eht(0, 0, 1) = 0: eht(0, 1, 1) = 3: eht(0, 2, 1) = 1: eht(0, 3, 1)
= 2
    eht(1, 0, 1) = 3: eht(1, 1, 1) = 0: eht(1, 2, 1) = 1: eht(1, 3, 1)
= 2
    eht(2, 0, 1) = 0: eht(2, 1, 1) = 1: eht(2, 2, 1) = 3: eht(2, 3, 1)
= 2
    eht(3, 0, 1) = 1: eht(3, 1, 1) = 0: eht(3, 2, 1) = 3: eht(3, 3, 1)
= 2

    Picture1.Print "Catch Dot Game Maze"

    px = 17 'プレイヤーキャラクタの初期座標
    py = 15
    modec = 0

    pm = 7 'プレイヤーキャラクタの通常の移動速度
    pm2 = 5 'プレイヤーキャラクタのスピードアップ中の移動速度
    pmc = pm

    dotc = 100

    '敵キャラクタ初期化
    For i = 0 To 3
        ef(i) = 1 '存在フラグをたてる
        eh(i) = 0
        em(i) = 9 + i * 2 '敵キャラクタによって速度差を付ける
        emc(i) = em(i)
    Next

    ex(0) = 1 '敵キャラクタの初期座標
    ey(0) = 5
    ex(1) = 33
    ey(1) = 5
    ex(2) = 1
    ey(2) = 25
    ex(3) = 33
    ey(3) = 25

    '最初に壁でゲームフィールドを埋める
    For y = 4 To 26
        For x = 0 To 34
            kg3(x, y) = 1 '壁に対応するコード
            Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15),
RGB(128, 0, 128), BF '壁表示
        Next
    Next

    x0 = 1 '基準点初期位置
    y0 = 5
    kg3(x0, y0) = 0 '空白
    Picture1.Line (x0 * 16, y0 * 16)-(x0 * 16 + 15, y0 * 16 + 15),
RGB(0, 0, 0), BF '空白（通路）表示
    spcc = 0 '起点データポインタ
    mef = 0

    While mef = 0
        t = 0
        For i = 0 To 3 '移動可能な方向を調べる
            If chkmazemove(x0, y0, i) > 0 Then
                hl(t) = i '移動可能な方向をh(t)にリストアップする
                t = t + 1
            End If
        Next
        If t > 0 Then '移動可能な方向があるか？
            p = Int(Rnd(1) * t) '乱数で移動可能な方向を選ぶ
            If t > 1 Then
                spx(spcc) = x0 '選ばれなかった方向が残る場合は起点デー
タとして保存
                spy(spcc) = y0
                spcc = spcc + 1
            End If
            x0 = x0 + dx(hl(p))
            y0 = y0 + dy(hl(p))
            kg3(x0, y0) = 0
            Picture1.Line (x0 * 16, y0 * 16)-(x0 * 16 + 15, y0 * 16 +
15), RGB(0, 0, 0), BF '空白（通路）表示
            x0 = x0 + dx(hl(p))
            y0 = y0 + dy(hl(p))
            Picture1.Line (x0 * 16, y0 * 16)-(x0 * 16 + 15, y0 * 16 +
15), RGB(0, 0, 0), BF '空白（通路）表示
            kg3(x0, y0) = 0
        Else
            If spcc < 1 Then
                mef = 1 '起点データが空になっていれば終了
            Else
                spcc = spcc - 1
                x0 = spx(spcc) '起点データから新たに掘り進む場所を取り
出す
                y0 = spy(spcc)
            End If
        End If
    Wend

    '外周は強制的に通路とする
    For y = 5 To 25 Step 20
        For x = 1 To 33
            kg3(x, y) = 0
            Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15),
RGB(0, 0, 0), BF
        Next
    Next
    For y = 5 To 25
        For x = 1 To 33 Step 32
            kg3(x, y) = 0
            Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15),
RGB(0, 0, 0), BF
        Next
    Next

    'プレイヤーキャラクタの出現場所の周囲は強制的に通路にする
    x = 17
    For y = 13 To 17
        kg3(x, y) = 0
        Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15),
RGB(0, 0, 0), BF '空白（通路）表示
    Next

    '仮想画面クリア
    For y = 4 To 26
        For x = 0 To 34
            kg1(x, y) = 0 '敵キャラクタが存在しないことを示すコード
            kg2(x, y) = 0 'スピードアップキャラクタが存在しないことを
示すコード
            For i = 0 To 3
                kg4(i, x, y) = 100000 '敵キャラクタ追跡用仮想画面クリ
ア
            Next
        Next
    Next

    'スピードアップキャラクタ初期化
    For i = 0 To 3
        sf(i) = 1 '存在フラグをたてる
        sh(i) = 0
        sm(i) = 10
        smc(i) = sm(i)
        '初期座標を乱数で決定
        x = Int(Rnd(1) * 35)
        y = Int(Rnd(1) * 23) + 4
        While (kg3(x, y) = 1 Or x = px Or y = py)
            x = x + 1 '壁やプレイヤーキャラクタと重なった場合は位置を
ずらしていく
            If x > 34 Then
                x = 1
                y = y + 1
                If y > 26 Then
                    y = 4
                End If
            End If
        Wend
        sx(i) = x
        sy(i) = y
    Next

    score = 0
    ti = 7000
    putscore

    gamestat = 0 'ゲームモード（０でゲーム中、１でゲームオーバー状態）

    Timer1.Interval = 10 'インターバルタイマ起動
End Sub

Private Function chkmazemove(x%, y%, h%)
    Dim xx%, yy%

    xx = x + dx(h) * 2
    yy = y + dy(h) * 2
    If xx < 1 Or yy < 1 Or xx > 33 Or yy > 25 Then
        chkmazemove = 0 'ゲームフィールド外部になってしまう場合
    Else
        If kg3(xx, yy) <> 1 Then
            chkmazemove = 0 'すでに掘ってある場合
        Else
            chkmazemove = 1 '壁の場合
        End If
    End If
End Function

Private Sub Picture1_MouseUp(button%, shift%, x!, y!)
    If (button And 2) = 2 Then 'マウス右ボタンクリックで強制終了
        End
    End If
End Sub

Private Sub Picture1_KeyDown(key%, shift%)
    If key = 27 Then 'ESCキーで強制終了
        End
    End If
    If key > 96 And key < 106 Then 'テンキー設定
        keystat(key - 96) = 1
    End If
    If key = 32 Then keytrig0 = 1 'トリガキー設定
    If key = 90 Then keytrig1 = 1
    If key = 88 Then keytrig2 = 1
    If key = 13 Then keytrig3 = 1
End Sub

Private Sub Picture1_KeyUp(key%, shift%)
    If key > 96 And key < 106 Then
       keystat(key - 96) = 0
    End If
    If key = 32 Then keytrig0 = 0
    If key = 90 Then keytrig1 = 0
    If key = 88 Then keytrig2 = 0
    If key = 13 Then keytrig3 = 0
End Sub

Private Sub Timer1_Timer()
    Dim npx%, npy%, npvx%, npvy%, nph1%, nph2%, mf%, nex%, ney%, nsx%,
nsy%
    Dim i%, f!, d%, h%, c%, k0%, k1%, nh%, i0%, i1%, i2%, x%, y%, m%
    Dim col!
    If gamestat = 0 Then '通常ゲームモード
        'プレイヤーキャラクタの処理
        pmc = pmc - 1 '移動カウンタ更新
        If pmc < 1 Then '移動するか？
            pmc = pm
            If spupc > 0 Then 'スピードアップ中は速度を上げる
                pmc = pm2
                spupc = spupc - 1
            End If
            If modec > 0 Then 'パワーアップ期間を数える
                modec = modec - 1
            End If
            npx = px
            npy = py
            npvx = 0
            npvy = 0
            If keystat(2) = 1 Then '下キーを押しているか？
                npvy = npvy + 1
            End If
            If keystat(4) = 1 Then '左キーを押しているか？
                npvx = npvx - 1
            End If
            If keystat(6) = 1 Then '右キーを押しているか？
                npvx = npvx + 1
            End If
            If keystat(8) = 1 Then '上キーを押しているか？
                npvy = npvy - 1
            End If
            If npvx <> 0 Or npvy <> 0 Then '移動キーを押しているか？
                If npvx <> 0 And npvy <> 0 Then '移動キーを２つ押して
いるか？
                    If ph = 0 Or ph = 2 Then
                        nph1 = 2 - npvx '第１候補
                        nph2 = 1 + npvy '第２候補
                    Else
                        nph1 = 1 + npvy
                        nph2 = 2 - npvx
                    End If
                Else
                    If npvx <> 0 Then
                        nph1 = 2 - npvx '押されている移動キーが１つの
場合は第１候補と第２候補を同じものにする
                        nph2 = 2 - npvx
                    Else
                        nph1 = 1 + npvy
                        nph2 = 1 + npvy
                    End If
                End If
                If kg3(npx + dx(nph1), npy + dy(nph1)) <> 1 Then '第１
候補の方向が壁で無いか？
                    mf = 1
                    ph = nph1
                ElseIf kg3(npx + dx(nph2), npy + dy(nph2)) <> 1 Then '
第２候補の方向が壁で無いか？
                    mf = 1
                    ph = nph2
                Else
                    mf = 0 '第１候補＆第２候補とも壁で移動できない場合
                End If
                If mf = 1 Then '移動できる場合
                    npx = npx + dx(ph) '移動できる方向に移動
                    npy = npy + dy(ph)
                    putplayer px, py, 0
                    putplayer npx, npy, RGB(255, 255, 255)
                    px = npx '座標更新
                    py = npy
                    If kg3(px, py) = 2 Then '移動先にドットがあるか？
                        kg3(px, py) = 0 '仮想画面のドットを消す
                        putdot px, py, 0
                        scoreup 10 + Int(ti / 100)    '得点アップ
                    End If
                    If kg3(px, py) = 3 Then '移動先にビッグドットがあ
るか？
```

→ つぎのページに続く

リスト1の続き

```
                        kg3(px, py) = 0 '仮想画面のビッグドットを消す
                        putbdot px, py, 0
                        scoreup 100
                        modec = modec + 30 + ti / 100
                    End If
                    If kg2(px, py) > 0 Then '移動先にスピードアップキャラクタがあるか？
                        sf(kg2(px, py) - 1) = 2 'キャラクタ消滅処理を指示
                        spupc = spupc + 50 '速度アップ期間追加
                    End If
                    If kg1(px, py) > 0 Then '移動先に敵キャラクタがあるか？
                        If ef(kg1(px, py) - 1) = 1 Then '通常状態の敵キャラクタか？
                            If modec > 0 Then 'パワーアップ中か？
                                ef(kg1(px, py) - 1) = 3  '敵キャラクタをしばらく消滅させる
                            Else
                                pf = 1 'プレイヤーキャラクタダメージフラグ
                                ef(kg1(px, py) - 1) = 2 '敵キャラクタをしばらく硬直させる
                                emc(kg1(px, py) - 1) = 100
                            End If
                        End If
                    End If
                End If
            End If
        End If
        If spupc > 0 Then col = RGB(255, 255, 0) Else col = RGB(255, 255, 255)
        If modec > 0 Then col = RGB(0, 255, 255)
        putplayer px, py, col
        '敵キャラクタの処理
        If modec > 0 Then
            col = RGB(0, 255, 0)
            m = 1
        Else
            col = RGB(255, 0, 0)
            m = 0
        End If
        For i = 0 To 3
            If ef(i) = 0 Then '存在していない場合
            ElseIf ef(i) = 1 Then '存在している場合
                emc(i) = emc(i) - 1 '移動カウンタ更新
                If emc(i) < 1 Then '移動するか？
                    emc(i) = em(i)
                    i0 = eh(i) 'インデックス１：敵キャラクタ方向
                    i1 = Sgn(px - ex(i)) + 1 'インデックス２：プレイヤーキャラクタとのＸ座標関係
                    i2 = Sgn(py - ey(i)) + 1 'インデックス３：プレイヤーキャラクタとのＹ座標関係
                    c = dp(i0, i1, i2) '位置関係テーブル参照
                    d = -1
                    For h = 0 To 2 '移動方向テーブルによる方向を順番にチェック
                        nh = eh(i) + eht(c, h, m)
                        If nh > 3 Then nh = nh - 4 '相対的方向のための補正
                        k0 = kg3(ex(i) + dx(nh), ey(i) + dy(nh))
                        k1 = kg1(ex(i) + dx(nh), ey(i) + dy(nh))
                        If (k0 = 0 Or k0 >= 2) And k1 = 0 Then     '移動先が空白かドットならば移動可能
                            If kg4(i, ex(i) + dx(nh), ey(i) + dy(nh)) - ti > 500 Or h = 3 Then '近い過去の間に通過しているかどうか？
                                d = nh
                                h = 2
                            End If
                        End If
                    Next
                    If d < 0 Then '移動履歴の影響で進行方向が見つからなかった場合は移動履歴を無視する
                        For h = 0 To 3 '移動方向テーブルによる方向を順番にチェック
                            nh = eh(i) + eht(c, h, m)
                            If nh > 3 Then nh = nh - 4 '相対的方向のための補正
                            k0 = kg3(ex(i) + dx(nh), ey(i) + dy(nh))
                            k1 = kg1(ex(i) + dx(nh), ey(i) + dy(nh))
                            If (k0 = 0 Or k0 >= 2) And k1 = 0 Then '移動先が空白かドットならば移動可能
                                d = nh
                                h = 3
                            End If
                        Next
                    End If
                    If d > -1 Then
                        eh(i) = d
                        nex = ex(i) + dx(eh(i)) '移動先の座標
                        ney = ey(i) + dy(eh(i))
                        putenemy ex(i), ey(i), RGB(0, 0, 0)
                        putenemy nex, ney, col
                        kg1(ex(i), ey(i)) = 0
                        kg1(nex, ney) = i + 1
                        If kg3(ex(i), ey(i)) = 2 Then '元の場所にドットがあった場合
                            putdot ex(i), ey(i), RGB(255, 255, 0) 'ドットを描き直す
                        End If
                        If kg3(ex(i), ey(i)) = 3 Then '元の場所にビッグドットがあった場合
                            putbdot ex(i), ey(i), RGB(255, 255, 0) 'ドットを描き直す
                        End If
                        If dotc < 1 Then '一定間隔でドットを発生させる
                            If Rnd(1) < 0.2 Then
                                putbdot ex(i), ey(i), RGB(255, 255, 0) 'ビッグドットを描く
                                kg3(ex(i), ey(i)) = 3 '仮想画面にビッグドットを設定
                            Else
                                putdot ex(i), ey(i), RGB(255, 255, 0) 'ドットを描く
                                kg3(ex(i), ey(i)) = 2 '仮想画面にドットを設定
                            End If
                            dotc = 100
                        End If
                        dotc = dotc - 1 'ドット発生カウンタ更新
                        ex(i) = nex
                        ey(i) = ney
                        kg4(i, nex, ney) = ti '残り時間を追跡用仮想画面に記録
                        If px = ex(i) And py = ey(i) Then 'プレイヤーキャラクタとの衝突判定
                            If modec > 0 Then
                                ef(i) = 3 '敵キャラクタをしばらく消滅させる
                            Else
                                pf = 1 'プレイヤーダメージフラグ
                                ef(i) = 2 '敵キャラクタをしばらく硬直させる
                                emc(i) = 100
                            End If
                        End If
                    End If
                End If
            ElseIf ef(i) = 2 Then '硬直している場合
                emc(i) = emc(i) - 1
                If emc(i) < 1 Then
                    ef(i) = 1 '通常に戻る
                End If
                putenemy ex(i), ey(i), RGB(128, 128, 128) 'スピードアップキャラクタなどと重なる場合があるので描画し直す
            ElseIf ef(i) = 3 Then '消滅開始
                kg1(ex(i), ey(i)) = 0 '仮想画面から消去
                scoreup 100 '得点アップ
                ef(i) = 4 '消滅状態へ移行
            ElseIf ef(i) = 4 Then '消滅中
                putenemy ex(i), ey(i), RGB(128, 128, 128)
                If modec < 1 Then 'パワーアップが終了したか？
                    ef(i) = 1 '通常状態へ復帰
                End If
            End If
        Next
        'スピードアップキャラクタの処理
        For i = 0 To 3
            If sf(i) = 0 Then '存在していない場合
            ElseIf sf(i) = 1 Then '存在している場合
                smc(i) = smc(i) - 1 '移動カウンタ更新
                If smc(i) < 1 Then '移動するか？
                    smc(i) = sm(i)
                    c = Int(Rnd(1) * 4) '乱数で移動方向優先順位を決める
                    d = -1
                    For h = 0 To 3 '移動方向テーブルによる方向を順番にチェック
                        nh = sh(i) + eht(c, h, 0)
                        If nh > 3 Then nh = nh - 4 '相対的方向のための補正
                        k0 = kg3(sx(i) + dx(nh), sy(i) + dy(nh))
                        k1 = kg2(sx(i) + dx(nh), sy(i) + dy(nh))
                        If (k0 = 0 Or k0 >= 2) And k1 = 0 Then     '移動先が空白かドットならば移動（敵キャラクタとは重なる場合あり）
                            d = nh
                            h = 3
                        End If
                    Next
                    If d > -1 Then
                        sh(i) = d
                        nsx = sx(i) + dx(sh(i)) '移動先の座標
                        nsy = sy(i) + dy(sh(i))
                        putspup sx(i), sy(i), RGB(0, 0, 0)
                        putspup nsx, nsy, RGB(0, 255, 0)
                        kg2(sx(i), sy(i)) = 0
                        kg2(nsx, nsy) = i + 1
                        If kg3(sx(i), sy(i)) = 2 And kg1(sx(i), sy(i)) = 0 Then '元の場所にドットだけがあった場合
                            putdot sx(i), sy(i), RGB(255, 255, 0) 'ドットを描き直す
                        End If
                        If kg3(sx(i), sy(i)) = 3 And kg1(sx(i), sy(i)) = 0 Then '元の場所にビッグドットだけがあった場合
                            putbdot sx(i), sy(i), RGB(255, 255, 0) 'ビッグドットを描き直す
                        End If
                        sx(i) = nsx
                        sy(i) = nsy
                        If px = sx(i) And py = sy(i) Then 'プレイヤーキャラクタとの衝突判定
                            spupc = spupc + 50 '速度アップ期間追加
                            sf(i) = 2
                        End If
                    End If
                End If
            ElseIf sf(i) = 2 Then 'プレイヤーキャラクタと衝突した場合
                kg2(sx(i), sy(i)) = 0
                putspup sx(i), sy(i), 0
                '新しい座標を乱数で決定
                x = Int(Rnd(1) * 35)
                y = Int(Rnd(1) * 23) + 4
                While (kg3(x, y) = 1 Or x = px Or y = py)
                    x = x + 1 '壁やプレイヤーキャラクタと重なった場合は位置をずらしていく
                    If x > 34 Then
                        x = 1
                        y = y + 1
                        If y > 26 Then
                            y = 4
                        End If
                    End If
                Wend
                sx(i) = x
                sy(i) = y
                sf(i) = 1
            End If
        Next
        If pf > 0 Then 'プレイヤーがダメージを受けている？
            ti = ti - 200 '残り時間を減らす
            pf = 0
        End If
        ti = ti - 1
        If ti < 0 Then ti = 0
        puttime
        If ti < 1 Then gamestat = 1 '時間切れでゲームオーバー
    ElseIf gamestat = 1 Then
        With Picture1
            .CurrentX = 230
            .CurrentY = 200
            .FontSize = 15
            .ForeColor = RGB(Rnd * 255, Rnd * 255, Rnd * 255)
        End With
        Picture1.Print "Game Over"
        If keytrig3 = 1 Then
            Timer1.Interval = 0 'インターバルタイマ停止
            gameinit
        End If
    End If
End Sub

Private Sub putscore()
    Picture1.Line (200, 0)-(369, 15), 0, BF
    With Picture1
        .CurrentX = 200
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "High Score " & hiscore
    Picture1.Line (370, 0)-(469, 15), 0, BF
    With Picture1
        .CurrentX = 370
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Score " & score
End Sub

Private Sub puttime()
    Picture1.Line (470, 0)-(575, 15), 0, BF
    With Picture1
        .CurrentX = 470
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Time " & ti
End Sub

Private Sub scoreup(up)
    score = score + up
    If hiscore < score Then hiscore = score
    putscore
End Sub

Private Sub putplayer(x, y, c)
    Picture1.Circle (x * 16 + 8, y * 16 + 8), 7, c
End Sub

Private Sub putdot(x, y, c)
    Picture1.Line (x * 16 + 7, y * 16 + 7)-(x * 16 + 8, y * 16 + 8), c, BF
End Sub

Private Sub putenemy(x, y, c)
    Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15), c
    Picture1.Line (x * 16 + 15, y * 16)-(x * 16, y * 16 + 15), c
End Sub

Private Sub putspup(x, y, c)
    Picture1.Line (x * 16 + 2, y * 16 + 2)-(x * 16 + 13, y * 16 + 13), c, B
End Sub

Private Sub putbdot(x, y, c)
    Picture1.Line (x * 16 + 5, y * 16 + 5)-(x * 16 + 10, y * 16 + 10), c, BF
End Sub
```

★本ページのプログラムはhttp://www.basicmagazine.netからダウンロードできます。

リスト1入力&実行方法

①VB起動

まずは，VBを起動します。新しいプロジェクトとして，[標準EXE]を選択して[開く]ボタンをクリックしてください。通常は，最初に標準EXEが選択された状態になっているので，[開く]をクリックするだけです。すると，**写真①**のような画面が表示されます。

②フォームの設定

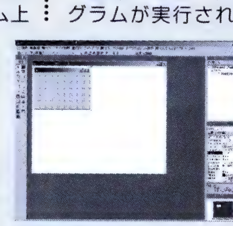
写真②のように，フォームを設定してください。プロパティは，プログラム中で更新されるので設定不要です（不用意に設定すると動作しなくなる場合があるので注意）。設定するのは，PictureBoxとTimerです。それぞれツールボックスのアイコンを選択した後に，フォーム上に貼り付けてください。貼り付ける操作は，フォーム上で左クリックした後，右下方向に少しドラッグしてから左ボタンを離せばOKです。位置，サイズなどは自由です（プログラム中で修正される）。

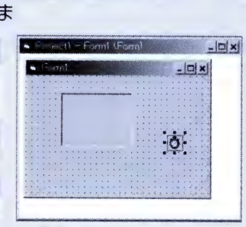
③リストの入力

プロジェクト・ウィンドウのコード表示ボタンをクリックすると，コード・ウィンドウが表示されます。Form1のコードとして，**リスト1**を入力してください。

④実行

実行メニューから[開始]を選択すると，プログラムが実行されます


《写真①》VBを起動したら「標準EXE」を選ぼう


《写真②》PictureBoxとTimerをフォームに配置する



はじめよう Microsoft Excelで BASICプログラミング

協力 富士通南九州システムエンジニアリング

第11回 コンピュータ対戦型のパズルゲームを作ろう(4)

この連載では、Microsoft Excelを使ってBASICのプログラミングを行います。今回は前回作成したオセロゲームを進化させます。また，先読みのテクニックを紹介します

TEXT & PROGRAMING 椛島たけし



3月は別れの季節。ノスタルジー(郷愁)を感じてしまう。そんな時，お勧めの映画が「ニューシネマパラダイス」。そういえばもうすぐ中学卒業10100周年同窓会があるな(もち2進)。

オセロ・ゲームを進化させる

さて、前回は「コンピュータ対戦型オセロゲーム」を何とか対戦ができる形まで作成しました(**第1図**)。コンピュータの手はランダムに決まりますので，当然強くはありません。ただ，このゲームを基本形にして，より強いゲームを作成することにしましょう。まずは，前回ヒントを出した方法で，改善してみることにします。

最も多く相手のコマを取れる場所

最も多く相手のコマを取れる場所を選ぶためには，コンピュータが置くことのできるマスの中で，最も相手のコマを取れる場所を調べる必要があります。実は，前回作成したランダムの場合のリストに仕掛けをしていますが気付いていますか？ コンピュータが打つことができるマスのデータは，ユーザ定義型の配列に格納していますが，このユーザ定義型配列のメンバに取れるコマの数を加えていたのです。したがって，**リスト1**の部分を置き換えだけで，最も多く相手のコマを取れる場所を選ぶことができます(ちなみに，最大数のマスが複数ある場合はランダムに決めています)。

《リスト１》 最も多く相手のコマを取れる場所を選ぶ

```
コンピュータの手を決める処理
Private Sub procCompIdea(IdeaCount As Long)
   Dim lngCompIdea As Long 'コンピュータが打てる手の場所数
   Dim lngWorkCtr As Long  'ワークカウンタ
   Dim lngMuchCtr As Long  '最も多い数
   Dim lngMuchIdea(64) As Long '最も多く返す手を格納する配列
   Dim lngMaxValue As Long '返すコマの最大数

   lngMaxValue = 0
   lngMuchCtr = 1
   lngMuchIdea(1) = 0

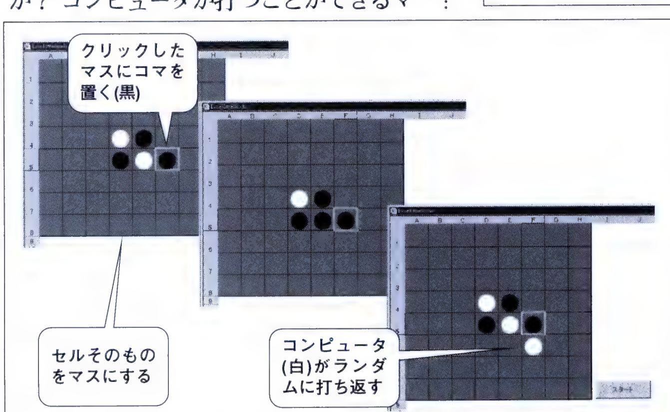
   For lngWorkCtr = 1 To IdeaCount '手の数だけループ
      Select Case udtIdea(lngWorkCtr).Rev '取れるコマの数
      Case Is = lngMaxValue   '最も多い数と同じ場合
         lngMuchCtr = lngMuchCtr + 1  '同数カウンタをカウント
         lngMuchIdea(lngMuchCtr) = lngWorkCtr '手の番号を格納
      Case Is > lngMaxValue   '最も多い数の場合
         lngMuchCtr = 1      '同数カウンタをリセット
         lngMaxValue = udtIdea(lngWorkCtr).Rev '最大数を保存
         lngMuchIdea(1) = lngWorkCtr '手の番号を格納
      Case Else
      End Select
   Next lngWorkCtr


   '最も多く返す手が複数ある場合ランダムに手を選択する
   lngCompIdea = lngMuchIdea(Int((lngMuchCtr * Rnd) + 1))

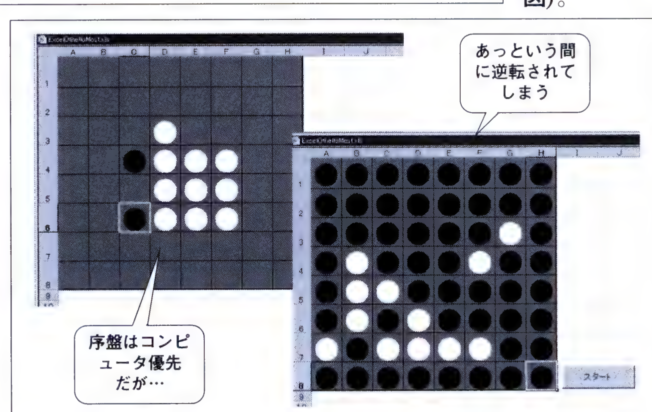
   'コマを返す処理
   Call procReverse(plngCompColor, udtIdea(lngCompIdea).Row, udtIdea(lngCompIdea).Col)
   Call procDrawPiece '盤面を描く

End Sub
```

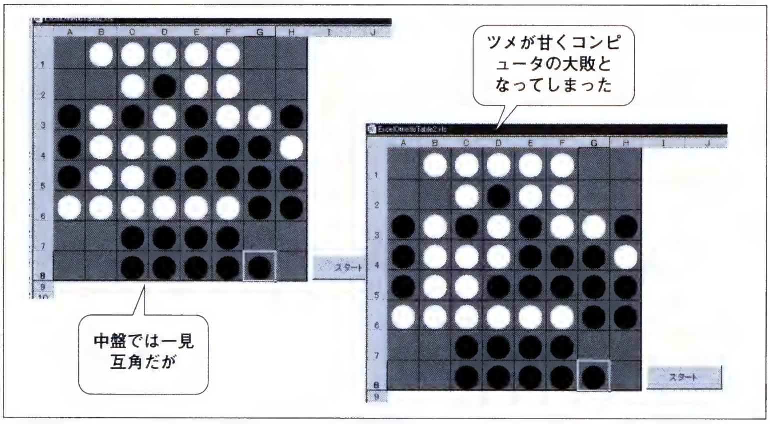
実際に対戦してみると，序盤ではコンピュータ優勢ですが，セオリーを無視した手がたたって，中盤～終盤でプレイヤーに簡単に逆転されるケースが多くなっています(**第2図**)。



《第1図》 前回はコンピュータが打ち返す部分まで完成した



《第2図》 常に最も多くのコマをとるマスを選ぶようにすると



第3図　ポイント表方式だと少し強くなる？

一般的に有利なマスを優先する

オセロのセオリーの第一歩は隅が有利ということでしょう。隅のように，一般的に有利なマスと不利なマス(隅の周囲など)を上手く使うことで，コンピュータを強くすることができます。

まず，**表1**のような，8×8のマスに一致させたポイント表を作ります。この場合，ポイントが高いほど優先順位が高くなります。したがって，隅のポイントは高く，隅の周囲のポイントは低く設定しています。このポイント表を，配列に読み取っておき，コンピュータの手を決める時にその配列を見ながら最も有効と思われる手を決定するわけです。少しリストが長くなりますが，効果が高いのでぜひ試してください(**リスト2**：このリストは前回のリストに加え得点計算処理などを加えています)。

実際に対戦すると，プレイヤーが負けることはほとんどありませんが，2〜3回プレイヤーがミスをするとコンピュータが勝つこともあります。

また，序盤，中盤，終盤でポイント表を変えてみるとコンピュータがもう少し強くなるはずです(**表2，リスト3**)。私の周りでも負ける人もいました(笑)。このポイント表を書き換えることで，少しは強くなりそうです。ただし，とくに終盤のツメの部分で変な手を選んで，コンピュータが勝ちを逃すケースもあり(**第3図**)，ポイント表だけでは「強い」レベルに達することはできません。

先を読むことを考えよう

さて，色々な方法で，コンピュータを強くすることを試しましたが，まだまだコンピュータが「強い」レベルではありません。本格的な強化のために，先読みについて考えてみましょう。

《表1》マスのポイント表

8	2	7	6	6	7	2	8
2	1	4	3	3	4	1	2
7	4	4	5	5	4	4	7
6	3	5	0	0	5	3	6
6	3	5	0	0	5	3	6
7	4	4	5	5	4	4	7
2	1	4	3	3	4	1	2
8	2	7	6	6	7	2	8

《表2》序盤・中盤・終盤でポイント表を変える

8	2	4	3	3	4	2	8
2	1	5	5	5	5	1	2
4	5	6	5	5	6	5	4
3	5	5	0	0	5	5	3
3	5	5	0	0	5	5	3
4	5	6	5	5	6	5	4
2	1	5	5	5	5	1	2
8	2	4	3	3	4	2	8

序盤用

8	2	7	6	6	7	2	8
2	1	3	4	4	3	1	2
7	3	4	5	5	4	4	7
6	4	5	0	0	5	4	6
6	4	5	0	0	5	4	6
7	3	4	5	5	4	4	7
2	1	3	4	4	3	1	2
8	2	7	6	6	7	2	8

中盤用

8	5	7	7	7	7	5	8
5	4	6	6	6	6	4	5
7	6	6	6	6	6	6	7
7	6	6	0	0	6	6	6
7	6	6	0	0	6	6	6
7	6	6	6	6	6	6	7
5	4	6	6	6	6	4	5
8	5	7	7	7	7	5	8

終盤用

《リスト2》前号リストに得点計算などを加えたリスト

```
Option Explicit

Private Const PIECE_CHR = "●" 'コマを表す文字
Private Const COL_BLACK = &H0     'コマの色(黒)
Private Const COL_WHITE = &HFFFF   'コマの色(白)
Private Const VAL_BLACK = -1  '黒の値
Private Const VAL_WHITE = 1   '白の値
Private Const VAL_SPACE = 0   '空の値

Private Const BOARD_TOP = 1    '盤の上位置
Private Const BOARD_LEFT = 1   '盤の左位置
Private Const BOARD_SIZE = 8   '盤のサイズ

Private Const ERR_OUTOF = 2&   'エラー(盤外)
Private Const ERR_SAME = 4&    'エラー(同じマス)
Private Const ERR_MISS = 8&    'エラー(打てないマス)
'ポイント用データ
Private Const POINTTABLE_DATA = "82766728" & _
                                "21433412" & _
                                "74455447" & _
                                "63500536" & _
                                "63500536" & _
                                "74455447" & _
                                "21433412" & _
                                "82766728"



'コンピュータの手の格納用のユーザ定義型
Private Type typIdea
   Row As Long
   Col As Long
   Rev As Long
End Type

Dim BoardStatus(0 To 9, 0 To 9) As Long 'オセロ盤の状況保存用配列
Dim plngCompColor As Long  'コンピュータのコマの色
Dim plngPlayerColor As Long 'プレイヤーのコマの色
Dim udtIdea(60) As typIdea  '手格納用ユーザ定義配列
Dim plngTurnCount As Long  'ターン数カウント用配列
Dim PointTable(1 To 8, 1 To 8) As Long 'ポイント表用配列

'スタートボタンが押された時の処理
Private Sub cmdStart_Click()
   Dim lngRowCtr As Long
   Dim lngColCtr As Long

   plngPlayerColor = VAL_BLACK
   plngCompColor = VAL_WHITE
   plngTurnCount = 1

   '状態配列をクリアする
   For lngRowCtr = 1 To 8
      For lngColCtr = 1 To 8
         BoardStatus(lngRowCtr, lngColCtr) = 0
      Next lngColCtr
   Next lngRowCtr

   Call procSetPoinTable
   '初期のコマの配置を行う
   BoardStatus(4, 4) = VAL_WHITE
```

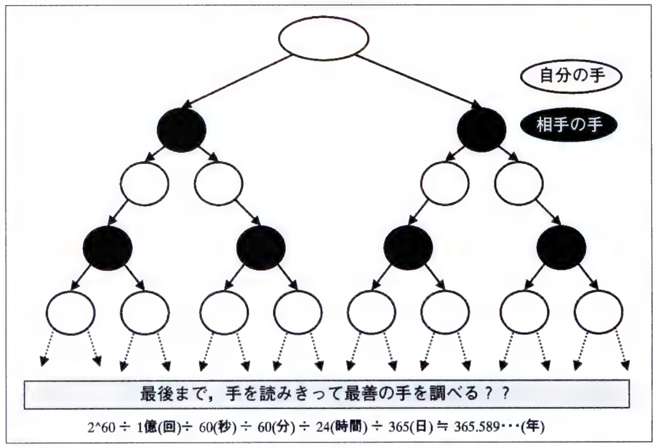

★本ページのプログラムはhttp://www.basicmagazine.netからダウンロードできます。

→つぎのページに続く

先読みの方法1

すべての手を読みきると……，先月の試算では数百年かかってしまいますね(**第4図**)。とりあえず，最後の3手を先読みすることを考えてみましょう。イメージすると**第5図**のようになります。まず，コンピュータの選択肢は3通りあります(空きマスが3つのため)。それに対するプレイヤーの手は2通りあります。したがって，最終的な手のパターンは6通りとなります。

最終的な結果(+8)のみを優先し，A3の手を選択すると，相手の手によっては最悪の結果(-3)になりかねません。そこで，相手は常に最良の手，つまり自分にとって最悪の手を常に選択すると考え，自分は最大の効果を選択することが必要です。このような考え方をミニ・マックス法と呼びます。ちなみに手の良し悪しを判断することを評価と呼びます。



第4図　可能性のある「手」をすべて判断しようとすると…

では，**第6図**を見ながら考えてみましょう。もしコンピュータがA1の手を選んだとします。プレーヤーは相手(コンピュータ)の結果が最も悪く

```
    BoardStatus(5, 5) = VAL_WHITE
    BoardStatus(4, 5) = VAL_BLACK
    BoardStatus(5, 4) = VAL_BLACK

    '画面表示処理を呼び出す
    Call procDrawPiece

    '一旦盤の外を選択する
    Cells(BOARD_TOP + BOARD_SIZE, BOARD_LEFT + BOARD_SIZE).Activate
End Sub

'ワークシートの選択セルが変更された時
'≒マウスがクリックされたとき
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Dim lngClickRow As Long 'クリックされた行
    Dim lngClickCol As Long 'クリックされた列
    Dim lngRetCode As Long 'プロシージャの戻り値
    Dim boolComp As Boolean
    Dim boolPlayer As Boolean

    Dim lngPlayer As Long 'プロシージャの戻り値
    Dim lngComp As Long 'プロシージャの戻り値

    Dim lngPlayerPoint As Long
    Dim lngCompPoint As Long
    Dim strResultMsg As String

    If plngTurnCount > 60 Then Exit Sub

    'プレイヤーのコマのチェック処理の呼び出し
    lngRetCode = procPlayerCheck(Target(1, 1).Row, Target(1, 1).Column)
    If lngRetCode = -1 Then '戻り値が-1(正しいマスを選択した場合)
        BoardStatus(Target(1, 1).Row, Target(1, 1).Column) = -1
        Call procDrawPiece 'コマを表示する処理

        'プレイヤーが打つ手がある間ループ
        Do Until lngPlayer > 0
            'コンピュータの打てる場所のチェック
            lngComp = procIdeaCheck(plngCompColor)
            If lngComp > 0 Then '打てる場所がある場合
                'コンピュータの手を決める処理
                Call procCompIdea(lngComp)
            End If
            'プレイヤーに打つ手があるかチェック
            lngPlayer = procIdeaCheck(plngPlayerColor)

            If lngPlayer = 0 And lngComp = 0 Then
                '得点計算処理
                Call procPointCalc(lngPlayerPoint, lngCompPoint)
                strResultMsg = "終了です。" & vbCrLf & _
                        "プレイヤー " & CStr(lngPlayerPoint) &
vbCrLf & _
                        "コンピュータ " & CStr(lngCompPoint)
                MsgBox strResultMsg
                Exit Do
            ElseIf lngPlayer = 0 Then
                MsgBox "プレイヤーは打つ手がありません。コンピュータの番で
す。"
            ElseIf lngComp = 0 Then
                MsgBox "コンピュータは打つ手がありません。あなたの番です。"
            Else
            End If
        Loop
```

```
    End If

End Sub

'コマを表示する処理
Private Sub procDrawPiece()
    Dim lngRowCtr As Long '行カウンタ
    Dim lngColCtr As Long '列カウンタ

    '配列のすべてを調べる
    For lngColCtr = 1 To 8
        For lngRowCtr = 1 To 8
            Select Case BoardStatus(lngRowCtr, lngColCtr)
            Case VAL_SPACE '空白なら
                Cells(lngRowCtr, lngColCtr).Value = ""
            Case VAL_BLACK '黒なら
                'セルの文字色に黒を設定し、●を表示する
                Cells(lngRowCtr, lngColCtr).Font.Color = COL_BLACK
                Cells(lngRowCtr, lngColCtr).Value = PIECE_CHR
            Case VAL_WHITE '白なら
                'セルの文字色に白を設定し、●を表示する
                Cells(lngRowCtr, lngColCtr).Font.Color = COL_WHITE
                Cells(lngRowCtr, lngColCtr).Value = PIECE_CHR
            End Select
        Next lngRowCtr
    Next lngColCtr

End Sub

'クリックされたマスについて調べる処理
Private Function procPlayerCheck(lngRow As Long, lngCol As Long) As Long
    Dim lngRowCtr As Long
    Dim lngColCtr As Long
    Dim lngReverse As Long
    Dim lngRetRev As Long
    Dim lngRetCode As Long

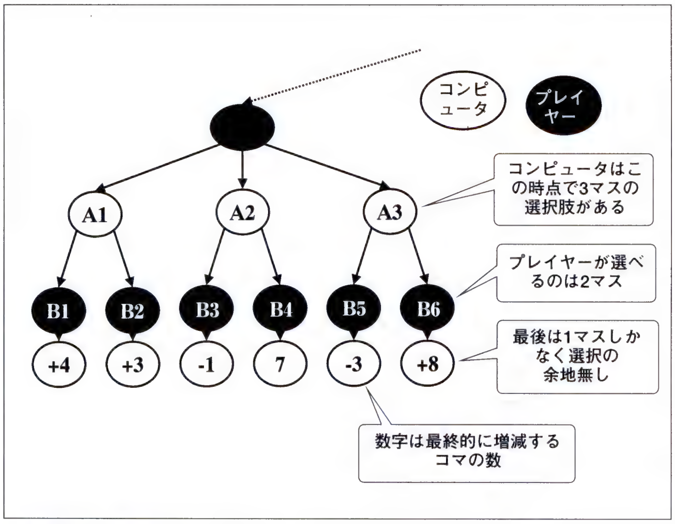
    'マス範囲外のチェック
    procPlayerCheck = 0
    Select Case lngRow
    Case BOARD_TOP To BOARD_TOP + BOARD_SIZE - 1
        Select Case lngCol
        Case BOARD_LEFT To BOARD_LEFT + BOARD_SIZE - 1
        'すでにコマが無いかチェック

            If BoardStatus(lngRow, lngCol) <> VAL_SPACE Then
                procPlayerCheck = ERR_SAME
            Else
                procPlayerCheck = -1
            End If
        Case Else
            procPlayerCheck = ERR_OUTOF
        End Select
    Case Else
        procPlayerCheck = ERR_OUTOF
    End Select

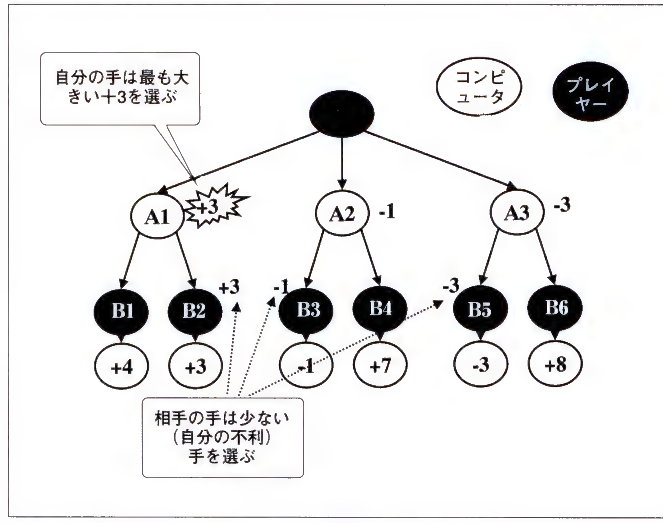
    If procPlayerCheck <> -1 Then Exit Function

    lngRetCode = procReverseCheck(plngPlayerColor, lngRow, lngCol)
    If lngRetCode > 0 Then
        Call procReverse(plngPlayerColor, lngRow, lngCol)
```

→ つぎのページに続く



《第5図》最後の3手のイメージ



《第6図》ミニ・マックス法を利用して手を決める

```
        procPlayerCheck = -1
    Else
        procPlayerCheck = ERR_MISS
    End If


End Function


'打つ手があるか調べる
Private Function procIdeaCheck(CheckColor As Long) As Long
    Dim lngRowCtr As Long '行カウンタ
    Dim lngColCtr As Long '列カウンタ
    Dim lngIdeaCtr As Long '手数のカウンタ
    Dim lngCompIdea As Long
    Dim lngRetCode As Long
    'コマを置ける位置の確認
    For lngRowCtr = 1 To 8
        For lngColCtr = 1 To 8
            '空白のマスの場合
            If BoardStatus(lngRowCtr, lngColCtr) = VAL_SPACE Then
                'コマを返せるか調べる
                lngRetCode = procReverseCheck(CheckColor, lngRowCtr,
lngColCtr)
                If lngRetCode > 0 Then
                    lngIdeaCtr = lngIdeaCtr + 1
                    With udtIdea(lngIdeaCtr)
                        .Row = lngRowCtr
                        .Col = lngColCtr
                        .Rev = lngRetCode
                    End With
                End If
            End If
        Next lngColCtr
    Next lngRowCtr

    procIdeaCheck = lngIdeaCtr

End Function
'コンピュータの手を決める処理
Private Sub procCompIdea(IdeaCount As Long)
    Dim lngCompIdea As Long
    Dim lngWorkCtr As Long
    Dim lngMuchCtr As Long
    Dim lngMuchIdea(64) As Long
    Dim lngMaxCount As Long

    lngMaxCount = 0
    lngMuchCtr = 1
    lngMuchIdea(1) = 0
'   PointTable(udtIdea(lngCompIdea).Row, udtIdea(lngCompIdea).Col)
    For lngWorkCtr = 1 To IdeaCount
        Select Case PointTable(udtIdea(lngWorkCtr).Row,
udtIdea(lngWorkCtr).Col)
        Case Is = lngMaxCount
            lngMuchCtr = lngMuchCtr + 1
            lngMuchIdea(lngMuchCtr) = lngWorkCtr
        Case Is > lngMaxCount
            lngMuchCtr = 1
            lngMaxCount = PointTable(udtIdea(lngWorkCtr).Row,
udtIdea(lngWorkCtr).Col)
            lngMuchIdea(1) = lngWorkCtr
        Case Else
```

```
        End Select
    Next lngWorkCtr

    lngCompIdea = lngMuchIdea(Int((lngMuchCtr * Rnd) + 1))

    'lngCompIdea = Int((IdeaCount * Rnd) + 1)

    Call procReverse(plngCompColor, udtIdea(lngCompIdea).Row,
udtIdea(lngCompIdea).Col)
    Call procDrawPiece

End Sub

'そのマスにコマを置けるか調べる
Private Function procReverseCheck(CheckColor As Long, NowRow As Long,
NowCol As Long) As Long
    Dim lngReverse As Long  '返すコマの総数
    Dim lngRetCode As Long  '返すコマの数

    lngReverse = 0
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, -1, -1) '左上
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, -1, 0) '上
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, -1, 1) '右上
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, 0, -1) '左
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, 0, 1) '右
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, 1, -1) '左下
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, 1, 0) '下
    lngReverse = lngReverse + lngRetCode
    lngRetCode = procCellsCheck(CheckColor, NowRow, NowCol, 1, 1) '右下
    lngReverse = lngReverse + lngRetCode

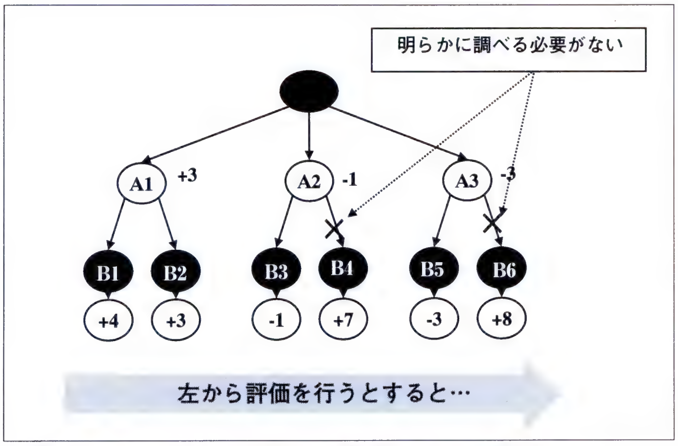
    procReverseCheck = lngReverse
End Function

'指定された方向について、返すコマの数を調べる
Private Function procCellsCheck(CheckColor As Long, _
                                NowRow As Long, _
                                NowCol As Long, _
                                RowStep As Long, _
                                ColStep As Long) As Long
    Dim lngTempRow As Long  '一時行カウンタ
    Dim lngTempCol As Long  '一時列カウンタ
    Dim lngRevCtr  As Long  '返せるコマの数
    Dim lngReverseValue As Long
    Dim lngSameValue As Long

    Select Case CheckColor  '調べる色の種類
    Case VAL_BLACK  '黒の場合
        lngSameValue = VAL_BLACK    '同じ色は黒
        lngReverseValue = VAL_WHITE '返す色は白
    Case VAL_WHITE  '白の場合
        lngSameValue = VAL_WHITE    '同じ色は白
        lngReverseValue = VAL_BLACK '返す色は黒
    End Select

    lngRevCtr = 0
    lngTempRow = NowRow
```

→ つぎのページに続く



《第7図》α-β法を利用して手を評価すると効率的になる

なる手B2を選択すると予想されます。したがってA1の評価値は+3となります。同様にA2の手を選んだ場合，プレイヤーの予想手はB4となりA2と評価値は-1，A3の評価値は-3となります。したがって，コンピュータ一の最善手は+3となるA1です。このように，「相手の手は評価値が最小のものを選び，自分の手は評価値が最大のものを選ぶ」のがミニ・マックス法なのです。ミニ・マックス法はいわばローリスク・ローリターンの堅実な方法と言えるでしょう。

先読みの方法2

ミニ・マックス法では最後の4手程度は，先読みが可能ですが，読む手数を増やすごとにネズミ算的に時間がかかります。コンピュータ対戦型のオセロや将棋，チェスなどが，ある意味で処理時間との戦いなのです。現実的な処理時間(コンピュータの計算時間)で，より多くの手を計算するには，不要な手をどれだけ減らせるかにかかっています。

α－β法はミニ・マックス法の応用編

```
    lngTempCol = NowCol

    Do
        lngTempRow = lngTempRow + RowStep
        lngTempCol = lngTempCol + ColStep
        Select Case BoardStatus(lngTempRow, lngTempCol)
        Case lngReverseValue '置くコマと違う色の場合
            lngRevCtr = lngRevCtr + 1
        Case lngSameValue  '置くコマと同じ色の場合
            If lngRevCtr > 0 Then  'カウントアップ
                procCellsCheck = lngRevCtr
            End If
            Exit Do
        Case VAL_SPACE  '空白の場合
            Exit Do
        End Select
    Loop

End Function


'指定されたマスにコマを置き、コマを返す
Private Sub procReverse(RevColor As Long, NowRow As Long, NowCol As Long)
    Dim lngTempRow As Long '一時行カウンタ
    Dim lngTempCol As Long '一時列カウンタ
    Dim lngTempCtr As Long '一時カウンタ
    Dim lngRevCtr  As Long '反転カウンタ
    Dim lngRowStep As Long '行方向
    Dim lngColStep As Long '列方向
    Dim lngSameValue As Long  '置いたコマと同じ色
    Dim lngReverseValue As Long '返すコマのいろ

    Select Case RevColor '色の種類
    Case VAL_BLACK '黒の場合
        lngSameValue = VAL_BLACK  '同じ色は黒
        lngReverseValue = VAL_WHITE '返す色は白
    Case VAL_WHITE '白の場合
        lngSameValue = VAL_WHITE  '同じ色は白
        lngReverseValue = VAL_BLACK '返す色は黒
    End Select


    '指定されたマスにコマを置く
    BoardStatus(NowRow, NowCol) = RevColor
    plngTurnCount = plngTurnCount + 1
    Call procDrawPiece

    If RevColor = plngPlayerColor Then
        MsgBox "プレイヤーの手は" & CStr(NowRow) & "行 " & CStr(NowCol) &
"列です。"
    Else
        MsgBox "コンピュータの手は" & CStr(NowRow) & "行 " & CStr(NowCol)
& "列です。"
    End If

    '8方向について返すコマを調べる
    For lngRowStep = -1 To 1
        For lngColStep = -1 To 1
            lngRevCtr = 0
            lngTempRow = NowRow
            lngTempCol = NowCol

            Do
                lngTempRow = lngTempRow + lngRowStep
                lngTempCol = lngTempCol + lngColStep
                Select Case BoardStatus(lngTempRow, lngTempCol)
                Case lngReverseValue '違う色の場合
                    lngRevCtr = lngRevCtr + 1
                Case lngSameValue  '同じ色の場合
                    If lngRevCtr > 0 Then  '返すコマがあったら
                        'コマを返す
                        For lngTempCtr = 1 To lngRevCtr
                            BoardStatus(NowRow + lngRowStep * lngTempCtr, _
                                NowCol + lngColStep * lngTempCtr) =
lngSameValue
                        Next lngTempCtr
                    End If
                    Exit Do
                Case VAL_SPACE '空白の場合
                    Exit Do
                End Select
            Loop
        Next lngColStep
    Next lngRowStep

End Sub


'得点計算処理
Private Sub procPointCalc(PlayerPoint As Long, CompPoint As Long)
    Dim lngRowCtr As Long
    Dim lngColCtr As Long

    plngPlayerColor = VAL_BLACK
    plngCompColor = VAL_WHITE

    '状態配列をクリアする
    For lngRowCtr = 1 To 8
        For lngColCtr = 1 To 8
            Select Case BoardStatus(lngRowCtr, lngColCtr)
            Case plngPlayerColor
                PlayerPoint = PlayerPoint + 1
            Case plngCompColor
                CompPoint = CompPoint + 1
            End Select
        Next lngColCtr
    Next lngRowCtr

End Sub

'ポイント表を読み込む処理
Private Sub procSetPoinTable()
    Dim lngRowCtr As Long '行用カウンタ
    Dim lngColCtr As Long '列用カウンタ
    Dim lngCellCtr As Long 'マスの位置

     For lngRowCtr = 1 To 8
        For lngColCtr = 1 To 8
            lngCellCtr = lngCellCtr + 1
            PointTable(lngRowCtr, lngColCtr) = _
                CLng(Mid(POINTTABLE_DATA, lngCellCtr, 1))
        Next lngColCtr
    Next lngRowCtr

End Sub
```

ですが，読む手数を減らすことができ，大幅に処理時間を短縮できます。

まず**第7図**を見てください。左側の手から手の評価を開始することにします。A1-B1とA1-B2のを調べA1の評価値が+3となります。続いて，A2-B3を調べた時点で，A2の評価値は-1以下となり，A1の評価値を上回ることが無いのがわかります。したがって，A2-B4の手は調べる必要がありません。同様にA3-B5の手を調べた段階でA3の評価値は-3以下となりA3-B6の手を調べません。

$\alpha-\beta$法ではこのように，先読みを行う必要が無い手を調べません。例題のように，読む手の数が少ない場合，ミニ・マックス法とあまり差がありませんが，手の数が増えるに従い威力を発揮します。ただし，手を最後まで読みきってしまう場合，$\alpha-\beta$法とミニ・マックス法で差はありませんので処理時間の早い$\alpha-\beta$法を利用するべきです。しかし，最後まで手を読みきれない中盤に，$\alpha-\beta$法を利用すると「最初は不利で最後に大逆転する手」を切り捨ててしまう可能性がありますので注意が必要です。

評価関数

例では最終的なコマの数をベースにミニ・マックス法や$\alpha-\beta$法評価を行いました。最後まで手を読むこととのできない中盤においても，これらの方法を利用するためには何らかの方法で，評価値を算出する必要があります。評価値を算出する関数を評価関数と呼びます。いわゆるセオリーを上手く数値化し，できるだけ有利な評価値を算出する「評価関数」を作成することがコンピュータ対戦型オセロの醍醐味かも知れませんね。評価関数の元となる比較的単純なセオリーには次のようなものがあります。

- 場所(隅有利，隅の周り不利)
- 相手が打てる場所の数(少ない方が良い)
- 自分が打てる場所の数(多い方が良い)

これらのセオリーを上手く取り込んで，評価関数を作ると，さらにコンピュータを強くすることができるはずです。

次回は，今回紹介した理論をプログラミングしコンピュータ対戦型オセロを完成させたいと思います。お楽しみに。

☆　☆　☆

《リスト3》 序盤・中盤・終盤でポイント表を変える

```
'宣言部分の追加リスト
'序盤ポイント表用データ
Private Const POINTTABLE_DATA1 = "82433428" & _
                        "21555512" & _
                        "45655654" & _
                        "35500553" & _
                        "35500553" & _
                        "45655654" & _
                        "21555512" & _
                        "82433428"

'中盤ポイント表用データ
Private Const POINTTABLE_DATA2 = "82766728" & _
                        "21344312" & _
                        "73455437" & _
                        "64500546" & _
                        "64500546" & _
                        "73455437" & _
                        "21344312" & _
                        "82766728"

'終盤ポイント表用データ
Private Const POINTTABLE_DATA3 = "85777758" & _
                        "54666645" & _
                        "76666667" & _
                        "76600667" & _
                        "76600667" & _
                        "76666667" & _
                        "54666645" & _
                        "85777758"

Dim PointTable1(1 To 8, 1 To 8) As Long 'ポイント表用配列(序盤)
Dim PointTable2(1 To 8, 1 To 8) As Long 'ポイント表用配列(中盤)
Dim PointTable3(1 To 8, 1 To 8) As Long 'ポイント表用配列(終盤)


'コンピュータの手を決める処理(変更)
Private Sub procCompIdea(IdeaCount As Long)
  Dim lngCompIdea As Long 'コンピュータが打てる手の場所数
  Dim lngWorkCtr As Long 'ワークカウンタ
  Dim lngMuchCtr As Long '高ポイント
  Dim lngMuchIdea(64) As Long '高ポイント格納用配列
  Dim lngMaxValue As Long '最高ポイント
  Dim lngPointValue As Long  'マスのポイント
  lngMaxValue = 0
  lngMuchCtr = 1
  lngMuchIdea(1) = 0

  For lngWorkCtr = 1 To IdeaCount
    Select Case plngTurnCount  'ターン数
    Case 1 To 22  '序盤
      lngPointValue = PointTable1(udtIdea(lngWorkCtr).Row, _
                              udtIdea(lngWorkCtr).Col)
    Case 23 To 40  '中盤
      lngPointValue = PointTable2(udtIdea(lngWorkCtr).Row, _
                              udtIdea(lngWorkCtr).Col)
    Case 41 To 60  '終盤
      lngPointValue = PointTable3(udtIdea(lngWorkCtr).Row, _
                              udtIdea(lngWorkCtr).Col)
    Case Else
    End Select

    Select Case lngPointValue  'マスのポイント
    Case Is = lngMaxValue '最大ポイントと同じ
      lngMuchCtr = lngMuchCtr + 1
      lngMuchIdea(lngMuchCtr) = lngWorkCtr '手の番号を格納
    Case Is > lngMaxValue '最大ポイントより大きい場合
      lngMuchCtr = 1
      lngMaxValue = lngPointValue '最大ポイントを保存
      lngMuchIdea(1) = lngWorkCtr '手の番号を格納
    Case Else
    End Select
  Next lngWorkCtr

  '最高ポイントの手が複数ある場合ランダムに手を選択する
  lngCompIdea = lngMuchIdea(Int((lngMuchCtr * Rnd) + 1))

  'コマを返す処理
  Call procReverse(plngCompColor, udtIdea(lngCompIdea).Row, udtIdea(lngCompIdea).Col)
  Call procDrawPiece '盤面を描く

End Sub


'ポイント表を読み込む処理(変更)
Private Sub procSetPoinTable()
  Dim lngRowCtr As Long  '行用カウンタ
  Dim lngColCtr As Long  '列用カウンタ
  Dim lngCellCtr As Long 'マスの位置

  For lngRowCtr = 1 To 8
    For lngColCtr = 1 To 8
      lngCellCtr = lngCellCtr + 1
      PointTable1(lngRowCtr, lngColCtr) = _
            CLng(Mid(POINTTABLE_DATA1, lngCellCtr, 1))
      PointTable2(lngRowCtr, lngColCtr) = _
            CLng(Mid(POINTTABLE_DATA2, lngCellCtr, 1))
      PointTable3(lngRowCtr, lngColCtr) = _
            CLng(Mid(POINTTABLE_DATA3, lngCellCtr, 1))
    Next lngColCtr
  Next lngRowCtr

End Sub
```

21世紀の遊びはコレ! ロボ道場

C言語でBN-1をプログラミング!!

取材/文　松原拓也

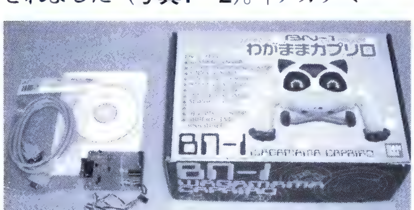
BN-1に本格的なプログラム開発キットが登場

BN-1 わがままカプリロ
アカデミーパック
C言語開発キット



「アカデミーパック」とは

「ロボットのプログラミングを勉強したい」という学生や研究者を対象に，ペットロボット「BN-1わがままカプリロ」とプログラム開発環境を組み合わせた「アカデミーパック」というセットが発売されました（**写真1〜2**）。「アカデミーパック」には「A・B・C」3種類のセットがありますが，中でも注目はBセット（またはCセット）に付属する「C言語開発キット」です。



▲《写真1》アカデミーパック（Bセット）



◀《写真2》バンダイ「BN-1 わがままカプリロ」。三つのサーボと二つの車輪，赤外線対物センサ，肉球（接触）センサ，ロータリーセンサ，なでなで（CdS）センサ，明るさセンサ，傾斜センサ，光フェロモン（赤外線信号）センサ，ジェスチャーセンサ（赤外線反射型），グラフィックアイ（8×8ドット3色），サウンド機能（4和音FM音源）を搭載

先に登場した「プログラミングキット（2001年9月号のロボ道場を参照）」は，専門的な知識を必要とせずに誰でもBN-1のプログラミングができるという，初心者〜中級者向けの開発キットでしたが，この「C言語開発キット」はC言語を使って，直接BN-1の制御ができる中級者〜上級者向けの開発キットです（対応機種はWindows，TurboLinux，FreeBSDです）。

「C言語開発キット」の中身

キットの中身は以下のようになっています（**写真3**）。

●「CD-ROM」

開発用のアプリケーションやサンプルプログラムやドキュメントが収録されています。

●「インターフェースボード+フラットケーブル」

パソコンとBN-1とで通信を行うためのハードウエアです。基板がむき出しです。

●「インターフェースユニット接続シリアルケーブル」

シリアル通信用のケーブル（ストレート）です。



▲《写真3》「C言語開発キット」。右側の基板がインターフェースボード

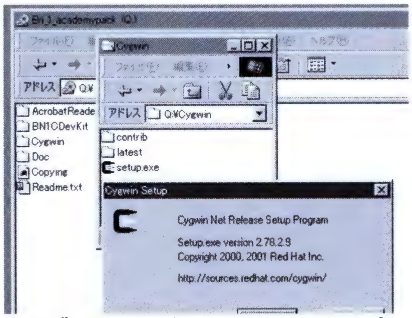
●「書類」

動作原理

BN-1は「H8/3067F」という，H8シリーズ（AKI-H8でおなじみ）のCPUを搭載しているため，「h8300-hms-gcc」というH8用のCコンパイラを使用します。Windowsの場合，「h8300-hms-gcc」は「cygwin」という動作環境で実行します。「cygwin」の操作画面は，グラフィカルユーザーインターフェース（GUI）を使わないUNIXのターミナルなでです。最初，操作に戸惑うかもしれませんが，ここでUNIX系のコマンドに慣れておきましょう。

BN-1には，ROM領域とSRAM領域というメモリ空間があり，C言語開発キッ

「C言語開発キット」使用法



▲まず，Cygwinをインストールします。CD-ROMの“Cygwin”フォルダにある“setup.exe”を実行します。インストールが完了するとパソコン上に“cygwin”フォルダが作られます。



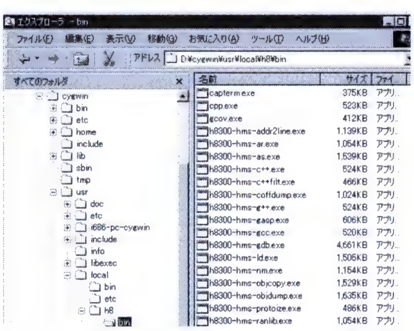
▲CD-ROMドライブの“BN1CDevKit”ディレクトリに移動して，BN-1用の開発環境のインストールします。Cygwinを実行して，以下のコマンドを入力します。

$ cd /cygdrive/q/BN1CDevKit [ENTER]
$ sh install.sh [ENTER]

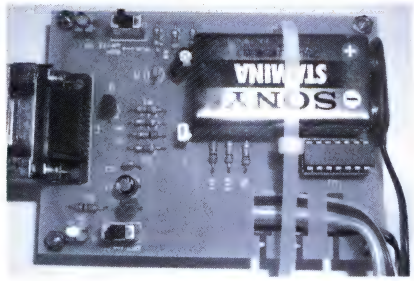


▲PATHを設定します。

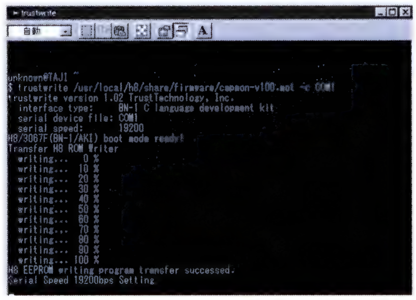
$ cp /usr/local/h8/share/temlate/bashrc.default ~/.bashrc [ENTER]
$ ls -a ~ [ENTER]
$ source ~/.bashrc [ENTER]



▲これで“/usr/local/h8/bin”にあるアプリケーション（Cコンパイラなど）を簡単に実行できるようになりました。



▲インターフェースボードの切り替えスイッチを「ROM WRITE」にセットして，電源スイッチをONにします。



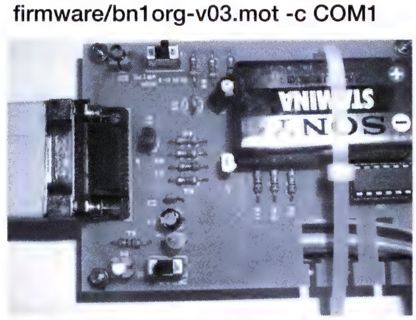
▲BN-1のROM領域を更新します。以下のコマンドでパソコンからファームウェアを転送します。

$trustwrite /usr/local/h8/share/firmware/capmon-v100.mot -c COM1



▲これでBN-1がC言語開発用になりました。逆に以下のコマンドでペットモードに戻すことができます。

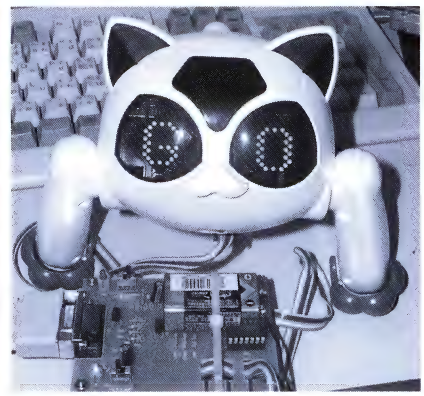
$ trustwrite /usr/local/h8/share/firmware/bn1org-v03.mot -c COM1





▲「カプリロモニター」を起動します。

$ capterm -c COM1 [ENTER]
そして，SRAM領域にサンプルプログラム（testsound.mot)を転送します。
load /usr/local/h8/share/samples/misc/testsound.mot

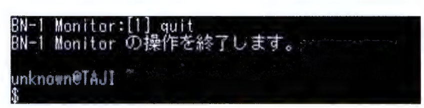


▲プログラムを実行します。このサンプルの場合，BN-1から音が鳴り出して，グラフィックアイに値が表示されます。

run [ENTER]



▲実行中，処理を一時停止させたり，値をパソコン側に表示させたりすることができます。



▲プログラムの動作が終わったら「カプリロモニター」を終了させます。

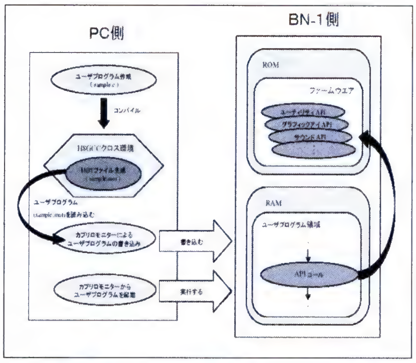
quit [ENTER]

◀インターフェースボードの切替スイッチを「RUN」にセットし，電源スイッチをONにします（元からONの場合はそのまま）。

ト用のファームウエアはROM領域に保存されます。センサを読み取ったり，モータを回したりといった，BN-1の制御に必要な関数（BN-1 API）もROM領域に保存されます。これらの関数はユーザーが作成したプログラムによって呼び出されます。ユーザーのプログラムはSRAM領域に保存されます（**写真4**）。

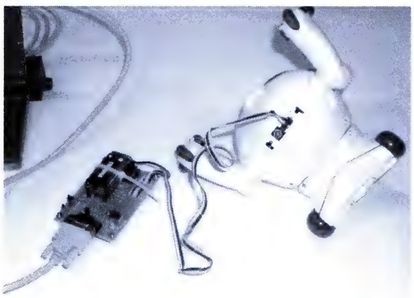
パソコンとの接続

まず，BN-1とパソコンを接続してみま



▲《写真4》ROM領域とSRAM領域の説明図（ユーザーマニュアルより）

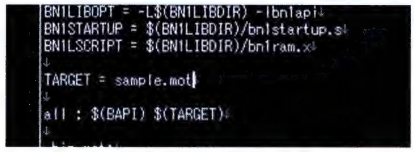
しょう（**写真5**）。パソコンとインターフェースボードをシリアルケーブル（D-sub9ピン）で接続。そして，インターフェースボードのフラットケーブル（6ピン）をBN-1に接続します。なお，インターフェースボードには9V乾電池（別売り）が必要です。



▲《写真5》パソコンとBN-1を接続した様子。BN-1側のコネクタはお腹の下



▲《写真6》作業用ディレクトリの作成



▲《写真7》MAKEFILEの書き換え

「C言語開発キット」使用法

開発環境の構築方法と，サンプルプログラムの実行方法です（Windows環境，シリアルポートCOM1の例です）。

プログラミング方法

では早速，オリジナルのプログラムを作って，BN-1を動かしてみましょう!!

下準備

最初に作業用として“~/bn1work”ディレクトリを作成します（**写真6**）。そして，“sample.c”というCのソースファイルと“Makefile”というMake用のファイルを作成します。Cのソースの書式は“/usr/local/h8/share/samples/misc”にあるサンプルが参考になります。Make用のファイルは“/usr/local/h8/share”ディレクトリにあるひな型をコピーしてから，エディタで「TERGET」の部分を“sample.mot”に変更しました（**写真7，リスト2**）。



▲《写真8》makeの実行。成功するとバイナリ（mot）ファイルが作られる

お手プログラム！

赤外線センサの特性を生かして，障害物の位置に合わせて腕が持ちあがるという，プログラムを作ってみました。名付けて「お手プログラム」です。

（リスト1の番号の解説です）

①BN-1制御用の関数「BN-1 API」を使

《リスト1》お手プログラム

```
/****************************************************/
/* BN-1 「お手プログラム」                    */
/*--------------------------------------------------*/
/* 左右の障害物に合わせてお手をします。       */
/****************************************************/

#include <bn1api.h>   /* BN-1 API ヘッダファイル */   ←←←①


/*---------*/
/* メイン */
/*---------*/
int main(void){     ←←←②

    SensorInfo_t sinfo;    /* センサ情報構造体 */
    char        eyebuf[3];
    int         i;

    /* モニターへ出力 */
    BN1TraceOut(1, "プログラム Start!");

    for(i=0 ;i<50 ;i++){
        BN1WaitMilliSec(200);
        BN1GetSensorInfo(&sinfo);   /* センサー情報を取得する */    ←←←③
        BN1CharToHexStr(eyebuf, sinfo.infrared);
        BN1TraceOut(2, eyebuf);
        BN1PutEye(eyebuf[1]); BN1PutEye(eyebuf[0]);    ←←←④

        if ( sinfo.infrared & (1 << 2) ) {
            BN1DoMotion(16); /*伏せ右腕あげ*/    ←←←⑤
        }
        if ( sinfo.infrared & (1 << 3) ) {
            BN1DoMotion(17); /*伏せ左腕あげ*/
        }
    }

    /* モニターへ出力 */
    BN1TraceOut(1, "赤外線センサー テストプログラム End!");

    return(0);
}
```

《リスト2》メイクファイル

```
#-----------------------------------------------
# Makefile
# BN1(H8/3067F) ユーザプログラムRAM実行環境用
#
# 2001/12/05 (c) Trust Technology, Inc.
#-----------------------------------------------

.SUFFIXES: .mot .bin .o .c

BN1INCDIR = /usr/local/h8/include/bn1
BN1LIBDIR = /usr/local/h8/lib/bn1

CFLAGS= -O -mh -I./ -I$(BN1INCDIR)
CC=h8300-hms-gcc

BN1LIBOPT = -L$(BN1LIBDIR) -lbn1api
BN1STARTUP = $(BN1LIBDIR)/bn1startup.s
BN1LSCRIPT = $(BN1LIBDIR)/bn1ram.x

TARGET = sample.mot

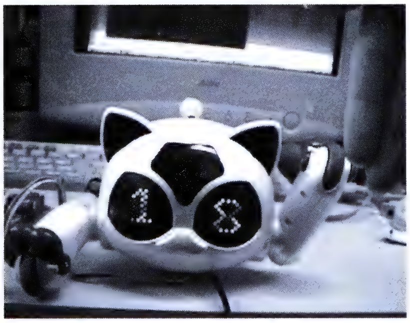
all : $(BAPI) $(TARGET)

.bin.mot:
	h8300-hms-objcopy -O srec $*.bin $*.mot
	chmod -x $*.mot
	rm -f $*.bin

.o.bin : $*.o rom3067.x 3067crt0.S $(BAPI) bn1api.h
	$(CC) $(CFLAGS) -T $(BN1LSCRIPT) -nostartfiles $(BN1STARTUP)
$*.o $(BN1LIBOPT) -o $@ -lc

.c.o:
	$(CC) $(CFLAGS) -o $*.o -c $*.c

clean :
	rm -f *.mot *.o
```



▲《写真9》左側に反応した状態（bit3が1に）。左腕が上がった！



▲《写真10》右側に反応した状態（bit2が1に）。右腕が上がった！

うためにヘッダファイル“bn1api.h”をインクルードします。
②処理はmain関数から始まります。プログラムは“for”ループで一定回数だけループして“return”命令で終了するようにしました。
③センサ情報構造体（**リスト3**）の“infrared”メンバーで赤外線センサの状態を知ることができます。“infrared”の内容は以下のとおりです。

bit0：前
bit1：後
bit2：右
bit3：左
bit4：下

その方向に障害物があるとそのビットが「1」になりますので，特定の障害物の有り/無しを検出できます。
④デバッグしやすいよう，センサの生データをBN-1の両目（16進数2桁）に表示させてみました。表示は“BN1PutEye”という関数を使っています。
⑤“BN1DoMotion”関数で腕を上げるモーションを行います（可能なモーションの数は約200）。単独でモータを制御することも可能です。

ビルド

makeを実行して，ソースのコンパイルを行い，バイナリ（拡張子はmot）を生成します。

```
$ make [ENTER]
```

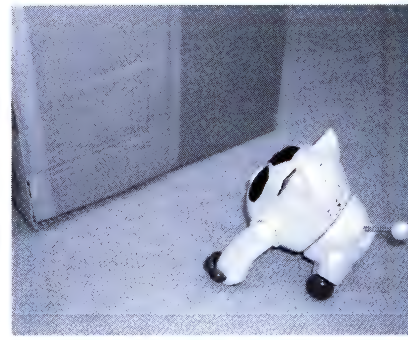
ビルドに成功すれば“sample.mot”が完成します（**写真8**）。

プログラムの転送

「カプリロモニター」で，“sample.mot”をBN-1に転送，そして実行します（ここは先に紹介した方法と同じです）。

```
$ capterm -c COM1 [ENTER]
 load sample.mot [ENTER]
 run [ENTER]
```

では，動作確認です。BN-1の左側に自分の手をかざしてみましょう。BN-1の左腕が上がりますね（**写真9**）。では，右側にかざすと……右腕が上がりました（**写真10**）。成功です！　BN-1が検知している障害物によって，グラフィックアイの表示内容が刻々と変化していくのも見ていて楽しいです。



▲《写真11》障害物を避けるプログラム

この機能を改良すれば，障害物を避けて自走するプログラムも簡単に作れます（**写真11**）。

最後に

このC言語のメリットを生かせば，高度な計算が必要な迷路脱出などのプログラムも作れるのではないでしょうか。C言語開発キットはWindowsの他，TurboLinuxやFreeBSDにも対応した本格的なロボット・プログラミングのプラットフォームとして，注目だと思います（他にはそうありません！）。

《リスト3》センサ情報構造体

```
/* センサー情報構造体 */
typedef struct {
    unsigned char  slant14;      /* 傾きセンサー１４方位          */
    unsigned char  slant6;       /* 傾きセンサー６方位            */
    Bool_t         overturnwarn; /* 転倒警告センサー              */
    unsigned char  infrared;     /* 赤外線センサー                */
    unsigned char  nikukyu;      /* 肉球センサー                  */
    Bool_t         shake;        /* シェイクセンサー              */
    unsigned char  chira1;       /* チラチラセンサー(前後左右)    */
    unsigned char  chira2;       /* チラチラセンサー(のど)        */
    unsigned char  mic;          /* マイクセンサー                */
    int            leftarm;      /* 左腕位置センサー              */
    int            rightarm;     /* 右腕位置センサー              */
    int            waist;        /* 腰位置センサー                */
    int            cds;          /* CDS(明るさ)センサー           */
    int            rotaly;       /* ロータリー(足)センサー        */
} SensorInfo_t;
```

自作一代記

電子工作にチャレンジ!



連載第11回
マイコンでBASICが動く!
「AKI-H8マイコンボード」

製作時間：約1日
制 作 費 ：110円
（マイコンボード代除く）
製作・文／松原拓也

今回は2001年12月号で紹介した秋月電子通商の「AKI-H8マイコンボード」をさらに使いこなしてみたいと思います。「マイコンボード」というのは組み込み用コンピュータを載せた汎用のボードのことです。

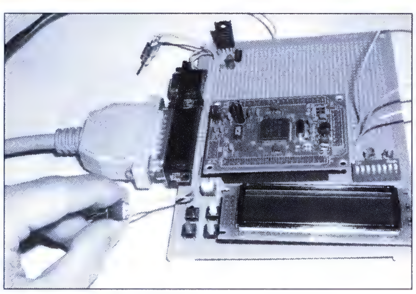
下の手順は，BASICコンパイラとアセンブラを使ってプログラムする方法です（対応機種はWindows98）。

A/Dとは

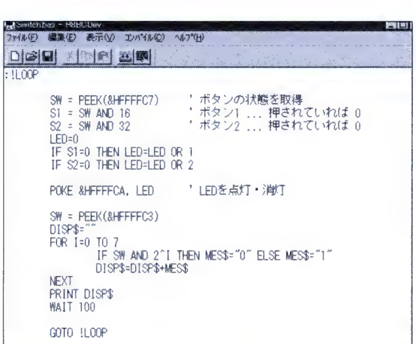
「AKI-H8マイコンボード」が内蔵している機能の一つにA/D変換があります。「A/D」とはアナログ（Analog）の入力をデジタル（Digital）に変換することを意味しますが，この場合，連続した電圧を量子化した値に置き換えて入力する機能です。

説明すると少々難しくなってしまうので，実際に製作してみましょう。以下の

AKI-H8プログラム方法



1）下準備としてBASICコンパイラとアセンブラをCD-ROMからハードディスクにコピーします。それから通信ケーブルでパソコン（シリアルポート）とマザーボード（25ピン）の間をつなぎます。ライターモードにして電源を入れれば準備完了です




```
:!LOOP

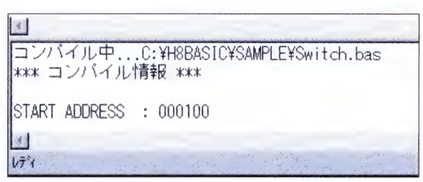
        SW = PEEK(&HFFFFC7)        ' ボタンの状態を取得
        S1 = SW AND 16             ' ボタン1 ... 押されていれば 0
        S2 = SW AND 32             ' ボタン2 ... 押されていれば 0
        LED=0
        IF S1=0 THEN LED=LED OR 1
        IF S2=0 THEN LED=LED OR 2

        POKE &HFFFFCA, LED         ' LEDを点灯・消灯

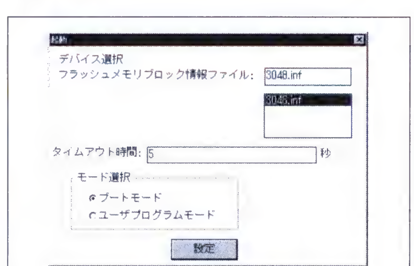
        SW = PEEK(&HFFFFC3)
        DISP$=""
        FOR I=0 TO 7
                IF SW AND 2^I THEN MES$="0" ELSE MES$="1"
                DISP$=DISP$+MES$
        NEXT
        PRINT DISP$
        WAIT 100

        GOTO !LOOP
```

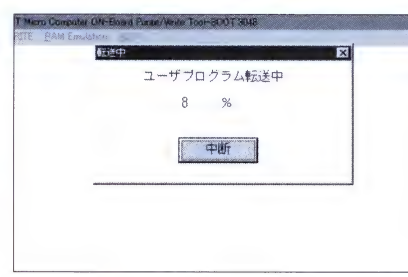
2）「BASICコンパイラ」でプログラムします。書式は「N88-BASIC」に似ていて，親しみやすいです。ファイルの拡張子は「BAS」で保存されます。



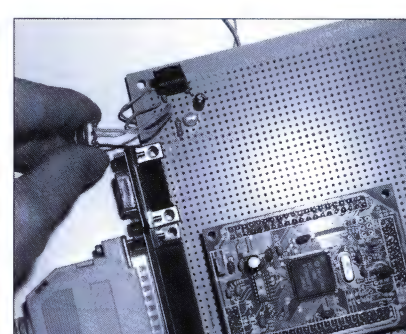
3）BASICのソースファイル（BAS）をコンパイルします。コンパイルに成功するとオブジェクトファイル（OBJ），ロードモジュールファイル（ABS），バイナリファイル（MOT）が自動的に生成されます。



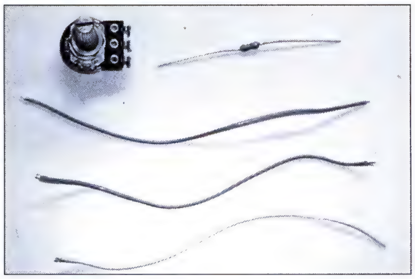
4）ROMライターソフト“FLASH.EXE”を使ってバイナリファイルをフラッシュROMに書き込みます。



5）バイナリファイルをフラッシュROMに書き込み中です。フラッシュROMは最低100回まで書き換えることが可能です（プログラムをRAM上にダウンロードすることも可能です）。



6）一旦電源を切り，通常モードにしてから，電源を入れるとプログラムが実行されます！



▲《写真1》必要なパーツ。可変抵抗，抵抗，リード線



▲《写真２》ボリューム式の可変抵抗100kΩ

パーツを用意してください（**写真1**）。

◆可変抵抗（約100kΩ，約50円）

ボリュームを回転させると抵抗値が変化する部品です（**写真2**）。3端子の可変抵抗の場合，1～3番間の抵抗値は変化せず，1～2番間と2～3番間の抵抗値の割合が変化します。

◆抵抗（約1kΩ，約5円）

一般的なカーボン抵抗です。

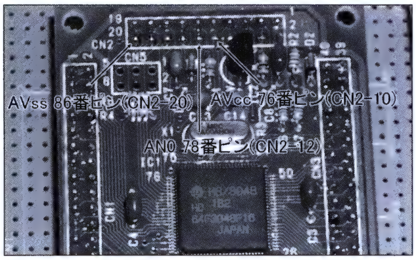
◆リード線（少々）

製 作

パーツをはんだ付けします（**写真3～5**）。はんだ付けするマザーボードの端子は以下のとおりです。

●「AVcc」76番ピン（CN2-10）

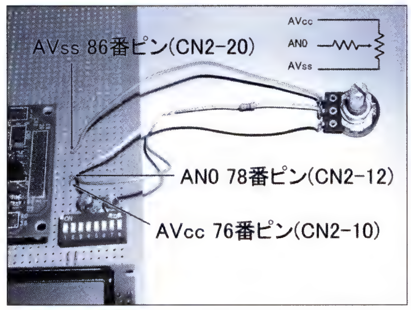
A/Dの上限となる基準電圧です。Vcc（5V）が接続されています。



▲《写真3》A/D変換を行うために必要なピン



▲《写真4》可変抵抗とマイコンボードをはんだ付け



▲《写真5》はんだ付けが完了

●「AN0」78番ピン（CN2-12）

A/Dを入力電圧です。AVssとAVccで基準電圧の範囲に限られます。

●「AVss」86番ピン（CN2-20）

A/Dの下限となる基準電圧です。Gnd（0V）が接続されています。

この他にも「AVref」という調整用の入力電圧がありますが，省略します。

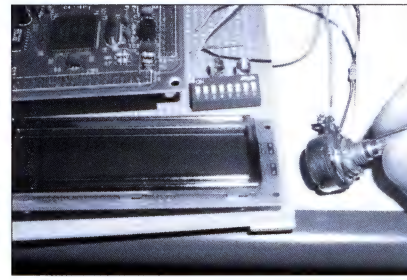
可変抵抗にはAVccから出ている5Vを分圧してA/D変換の入力端子に接続します。可変抵抗の2番端子の抵抗はボリュームを絞り過ぎて0Ωになってしまうことを防ぐものです。

プログラム

では，A/Dを行って液晶パネルに表示するサンプルプログラムを作ってみましょう。プログラムは標準で付属する「BASICコンパイラ」を使います。便利なことにA/D入力は「ADINP」という関数1個で行えますので，わずか5行で完成です。

```
:!LOOP
 ADATA = ADINP(0)
 PRINT STR$(ADATA)
 WAIT 100
 GOTO !LOOP
```

引数の「0」はA/Dのチャンネル番号（0～7）です。戻り値はA/Dされた数値



▲《写真6》プログラムの実行中。ボリュームの回転に合わせてA/Dの値が変化



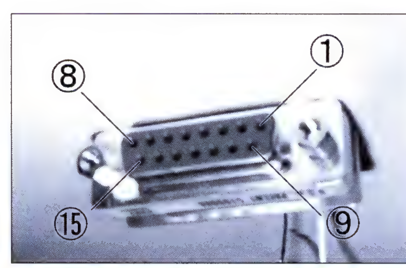
▲《写真7》ゲームポート仕様のジョイスティック（実売価格980円!?　もらい物なのでタダ）

（0～65535）を表しています。

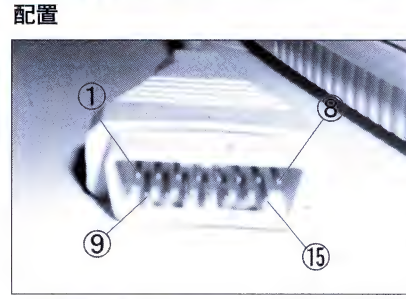
プログラムを実行すると，ボリュームの回転に合わせて液晶の数値が増減します。成功です!!（**写真6**）

PC用ジョイスティックを接続！

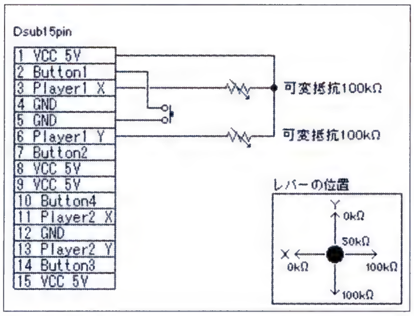
このままでは面白くないので，A/Dを使ったゲームを作ってみました。さらに，入力装置を可変抵抗のかわりにゲームポート（GamePort）仕様のパソコン用ジ



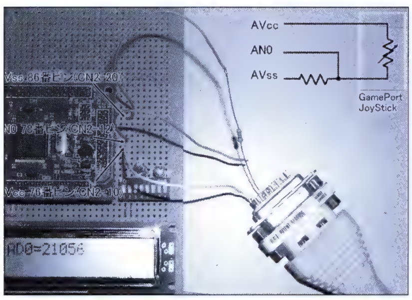
▲《写真8》Dsub15pin（メス）のピン配置



▲《写真9》Dsub15pin（オス）のピン配置

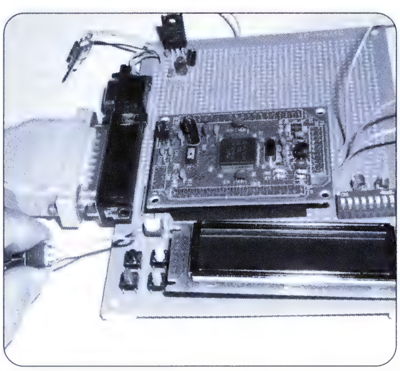


▲《写真10》ジョイスティックの等価回路。部品に100kΩの可変抵抗が使われている（十字キー型ではこの例に当てはまりません）



▲《写真11》ジョイスティック用コネクタとマイコンボードとをはんだ付け

ョイスティック（**写真7**）に交換してみたいと思います。



▲「AKI-H8/3048Fマイコンボード」7,800円

プロセッサは日立製の「H8/3048F」（16bitCPU，動作クロック16MHz。フラッシュROM容量128kバイト）。

マザーボードにはスイッチ（タクトスイッチ4個，8Pディップスイッチ1個）と液晶パネル（16文字×2行の液晶）とROMライタを搭載。

CD-ROM（アセンブラ，BASICコンパイラ）付属。

＊キットには電源（14V以上，200mA以上）と通信ケーブルが付属しませんので，別途用意しておきましょう。

◆秋月電子通商
http://www.akizuki.ne.jp/



▲《写真12》「数集めゲーム」を実行中。レバーの微妙な傾きが自機（Iマーク）の位置に反映される

なお，ゲームポート側のコネクタはDsub15pin（オス）なので，Dsub15pin（メス）を用意しましょう（約50円）。ゲームポートのピン配置は以下のとおりです（**写真8〜9**）。

1 VCC 5V
2 Button1
3 Player1 X
4 GND
5 GND

《リスト1》ソースプログラム **"joygame.bas"**

```
' A/D入力 サンプルゲーム
  ASTAT = ADINP(0)   'レバーの初期位置

  CONSOLE 0      '2行モードに
  SPC$="                " '空白16個
:!START
  LPCNT=100    'ループ回数
  SCORE=0      'スコア
  GOSUB !TEKIBONE      '敵の出現

:!LOOP
    JIKIX = ADINP(0)    'レバー入力
    JIKIX = 8-((JIKIX - ASTAT) / 1800)  '自機X座標

    IF JIKIX < 1  THEN JIKIX = 1
    IF JIKIX > 16 THEN JIKIX = 16

    OTI = OTI - 1  '敵の滞在時間カウンタ

    DISP$ = RIGHT$(SPC$,TEKIX-1) + CHR$(&H30 + OTI) + LEFT$(SPC$,16-TEKIX)
    DISP$ = DISP$ + RIGHT$(SPC$,JIKIX-1) + "|" + LEFT$(SPC$,16-JIKIX)
    PRINT DISP$   '画面表示
    WAIT 100

    IF TEKIX = JIKIX THEN SCORE = SCORE + OTI
    IF OTI = 0 THEN GOSUB !TEKIBONE

    LPCNT=LPCNT-1  'ループ回数を減算
    IF LPCNT > 0 THEN GOTO !LOOP

    PRINT "SCORE:";SCORE
    WAIT 4000
  GOTO !START
'---------敵出現
:!TEKIBONE
  OTI = 9
  TEKIX = RND(16)+1  '敵のX座標
  RETURN
```



▲《写真13》実行中の画面

6 Player1 Y
7 Button2
8 VCC 5V
9 VCC 5V
10 Button4
11 Player2 X
12 GND
13 Player2 Y
14 Button3
15 VCC 5V

内部の仕掛けは等価回路（**写真10**）を見てのとおりですが，ジョイスティックのレバーの位置はVCC～X（又はY）間の抵抗値となって反映されます。中間が50kΩ，右に倒すと100kΩ，左に倒すと0Ωです（つまりアナログ・ジョイスティック）。この特製を応用すれば，VCCの電圧を分圧して，AD0に入力することが可能です（**写真11**）。

完成したプログラムが**リスト1**です。さあ，うまく動くでしょうか？



▲《写真14》一定時間が経過すると得点表示

数集めゲーム

電源を入れて実行……レバーを動かすと，液晶に表示された自機（Iマーク）も連動して動きます。乱数で表示される敵とX座標を合わせると得点が加算されます。一定時間に達するとゲームオーバーです（**写真12～14**）。成功です。

ゲームポート仕様のジョイスティックは，すっかり見かけなくなってしまいましたが（中古でならまだ出まわっているかも），構造がシンプルなので電子工作への流用がしやすいみたいです。思わぬ発見でした。

最後に

AKI-H8には，今回紹介したA/Dの他にD/Aという逆の働きの機能があります(デジタルの値をアナログの電圧で表す)。まだまだ使いきっていない機能が残っていますので，機会があったらまた紹介したいと思ってます。

AKI-H8で使えるBASIC

「AKI-H8/3048マイコンボード」はBASIC言語でプログラムすることができます。

付属のBASICは，その場で即実行できるインタープリタ型とは違い，BASICコンパイラでバイナリに変換する作業（コンパイル）があらかじめ必要です。

BASIC言語の仕様はN88-BASICに似ていますので親しみやすいかもしれません。

BASICコンパイラの予約語一覧

予約語	END	MOD	READ	STRING$
ABS	FOR	NEXT	REM	THEN
ADINP	GOSUB	NOT	RESTORE	TIMER
AND	GOTO	ON	RETURN	TO
ASC	HEX$	OR	RETURNIRQ	TIMERSET
CALL	IF	PEEK	RIGHT$	VAL
CHR$	INPUT$	PEEKW	RND	VARPTR
CONSOLE	INSTR	POKE	SERIALSET	WAIT
CONTINUE	IRQ	POKEW	SPACE$	WHILE
DAOUT	LEFT$	PORTSET	START	WEND
DATA	LEN	PRINT	STEP	XOR
DIM	LET	RANDOMIZE	STOP	
ELSE	MID$		STR$	

あくせすROOM



読者参加お待ちします

影：ううー寒い！　このサイフの寒さを見るかぎり，日本の不景気はどん底だね。



▲東京都練馬区・ルストフレム
つぐ美：他のコーナーへの投稿もよろしくね！
影：そういえば，年賀状をまだ出してなかったっけ。投稿に流用するか。

つぐ美：影さんのサイフの軽さかげんは，昔からでしょうに。

編：影さんがどんどんパソコンを壊しちゃうから，ベーマガ編集部の経理も大変な状態なんですよ。

影：うおーそうだったのかー。ベーマガを救えプロジェクト発動。読者諸君，おもしろいハガキでさらなる読者を招き寄せて，発行部数大躍進で，この不況を乗り切るのだー。

編：そんなわけで，今月の「あくせすROOM」です。



◀大阪府吹田市・ARAI
影：かっちょえぇ～。刑事ドラマの1シーンみたい。
つぐ美：カゲのあるDr.Dってシブいー。
編集長：おお、ワシもいるじゃあないか。



◀埼玉県北足立郡・ぶんちょ
影：少々寒くても、木枯らし吹こうとも、若者たちは元気なのだ。
つぐ美：わたしだって負けてないわよ！　ほらナマ足とミニスカート！
編：見てるこっちが寒くなります。



◀愛知県名古屋市・わいち301
編：カラーイラストが、もったいないですね。
つぐ美：よく見ると、カルタの文章がすっごいヘンなの。

萌えプロ番外編

影：いまだに萌えプロ宛の投稿があるので，ワクを作ってみたぞ。お題はとくに設けないので，各自の判断で好き勝手に書いてほしい。



▶福島市・ショタ委員みなみ
つぐ美：そうよねー，男の子のイラストもあっていいわよねー。
編：いたいけな青少年が早くも毒牙にかけられようとしています。

うろおぼえ友の会

影：今月は「バールのようなもの」の謎に迫りたい。ニュースなどで耳にする，金庫などをこじ開けるのに使われる「バールのようなもの」。おそらくクギ抜きみたいな形をしているのだろうけど，なぜかハッキリと「バールだ！」とは誰も言い切らない。なぜ？　もしかしたら，本当にそういうものがあって，私が忘れているだけなのか？　というわけで，
今月の掲載は「バールのようなもの」

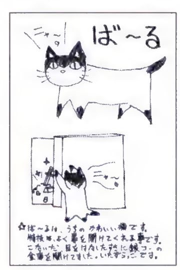
次回の募集

影：よっしゃー次の募集は「けさらんぱさらん」だ！　応募締め切りは3月15日！　図書館で調べないようにっ!!

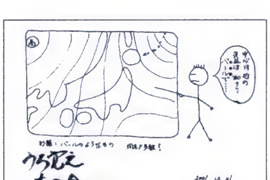
編：なんだか，もう「想像してみましょう」的な企画になってますね。そうそう，お題も送ってください。

▼福島市・南斗オレガ
影：世界中に名前の残っている古代神。そうか，名前を言うのも恐ろしくて「ようなもの」とか言っているのか！
編：もしかすると，セト神とかバアル・ゼブブとか，亜種が多いのかもしれませんよ。

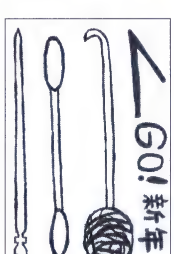




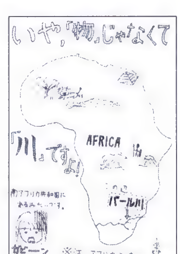
◀愛知県名古屋市・のうぎょうアッパー
影：なんとネコさんであったか！
編：似たようなネコが全国で大活躍してるんですね。



▲大阪府堺市・3代目ryo（10%増量中）
編：今ではヘクトパスカルという国際単位を使うようになっていますね。
Dr.D：わしは関節の痛み具合で，気圧がわかるぞ！
影：私は気圧の下がり具合で，花粉症みたいな症状が出るのだ！



▲愛知県名古屋市・最馬場可
影：これは……ゴメン，わからないや。本物のバール？
つぐ美：これはアレね！　海賊が腕とかにつけるやつよ。アタッチメントってやつ？
編：いや，それは……。



▲東京都練馬区・ルストフレム
影：九州かと思った。
編：アフリカって書いてあるじゃないですか。
影：九州県アフリカ市かと。
編：え，九州……県？

あくせすROOMの参加方法

あくせすROOMでは，みなさんの投稿をお待ちしています。愛読者カードや，官製ハガキ，または同サイズの丈夫な紙などを封書でまとめるなどして，お送りください。

鉛筆や色鉛筆はきれいに印刷されません。せっかくペンで描いてあっても，鉛筆の下書きが残っているものは，やはり載りにくくなりますので，インクや水性ボールペンなど，はっきりと書ける道具を使いましょう。文字だけの投稿であれば，ワープロを使ってもかまいませんよ（そのまま掲載できます）。

応募先
〒141-8715　東京都品川区東五反田1-11-15
電波新聞社マイコンBASICマガジン編集部
あくせすROOM係

《募集物の一例》

●うろ覚え選手権
聞いてもなんだかサッパリわからない「けさらんぱさらん」なる物体（そうか，これは物体なのか!?）を，資料を見ないで想像力だけをたよりに，イラストにしてみてください。3月15日必着です。

●世界のカスタム野郎
次回から，愛用の道具をいろいろカスタマイズ（自己改造）して使っている人の自慢話を募集します。冷蔵庫にパーツを組み込んでDOS／Vパソコンを作る！　ゲームコントローラに新しく機能をつける！　ちょっとした工夫で，実用性がアップしたり，カッコ良くなったという実例を，ハガキ1枚で紹介してください。

●デジカメ激写ボーイ
デジカメを持って街に出よう！　デジカメやデジビで撮影したユニークなお笑い画像を送ってください。

あくせすROOMは，みなさんのパワーで作るページです。このほかにも，おもしろいネタがあったら，文字でもイラストでも，思いつくままに表現してみてください。企画として長くつづくポイントは，ほかの人がちょっとしたアレンジで相乗りできること。みんなで楽しく参加できそうなネタを常時募集しています。

3月号お楽しみプレゼント

①「AIBO ERS-220」特性Tシャツ他／1名（提供:ソニー）
②携帯ストラップ各種／5名（提供：編集部）
③コーエー定番シリーズ／各5名（提供：コーエー）
③-1大航海時代Ⅱ
③-2太閤立志伝Ⅱ
③-3三國志孔明伝

●応募方法：本誌とじ込み愛読者ハガキ（またはインターネット読者ハガキ）にプレゼント記入欄があります。一つだけ選んでお送りください。

●締め切り：2002年3月7日必着

《2002年1月号の当選者》①京都府中郡・瀬川邦彦さん他2名　②埼玉県さいたま市・大熊拓也さん　③愛知県宝飯郡・渡辺慎吾さん他2名　④神奈川県横浜市・門脇友貴さん

Falcom® NEWS EXPRESS

日本ファルコム株式会社
〒190-0012 東京都立川市曙町1-14-13
TEL 042-527-6501
FAX 042-528-2714
スタッフ募集中
採用担当BFM係
http://www.falcom.co.jp/

「Zwei!!（ツヴァイ）」NEWパッケージは七瀬葵新作描きおろし!



PCゲームでは年末最大のヒットとなった「Zwei!!」。昨年12月中に完売したという店も続出し，なかなか入手できないという状況が続いていましたが，遂に七瀬葵新作描きおろしのNEWパッケージ版が登場！

今回のパッケージも一見すると普通のパッケージですが，何と中の内箱までフルカラー印刷にするなど，またまた凝りまくったパッケージとなっています。一旦は販売終了していたファルコムホームページでも，通販が再開していますので，お近くにお店がない方は便利な通信販売をぜひご利用ください。

http://www.falcom.co.jp/zwei/order.html

試してZwei!!キャンペーン実施中！

もっといろんな人にZwei!!を遊んでもらいたい！買おうかどうか悩んでいる人に試してもらってZwei!!の面白さを実感してもらいたい！ということで体験版の無料ダウンロードを実施中！すでに心ゆくまでZwei!!をプレイしたという方も，スタッフこだわりの体験版もぜひ遊んでみてね！

4Gamer Net
http://www.4gamer.net/news/history/2002.01/20020111200446detail.html

enban.net
http://enban.net/cgi-bin/view.cgi?file=game/taikenban-d/index.html

操作マニュアルは，ファルコムのホームページからダウンロードできますので，ぜひご利用ください。

http://www.falcom.co.jp/zwei/download.html

下記サイトでもダウンロード予定となっています。

●ゲームサイトEG　●毎日新聞社ゲームクエスト
●電撃オンライン　●オンラインプレイヤー

Zwei!!攻略ガイドは2/10発売！

Zwei!!はシナリオをクリアするだけでなく，いろんな楽しみが満載の奥の深いゲームです。モンスター辞典やアイテム辞典のコンプリート，レアアイテムが入手できる，わらしべイベントにミニゲームなど，隠された要素もいっぱいです。2周目の攻略もバッチリの攻略ガイドでZwei!!の完全制覇をめざそう！



発行：角川書店　予価：2,200円

BIGLOBEとパピレスでダウンロード販売開始！

ベクターでも好評発売中のファルコムのダウンロード版6タイトルが，BIGLOBEのSoft plazaとパピレスでも販売が開始されました。パッケージ版は入手が困難なタイトルもありますので，価格も安く気軽に購入できるダウンロード版をぜひ遊んでくださいね！

●ソーサリアンforever　●新英雄伝説　●ヴァンテージマスター　●ヴァンテージマスターオンライン　●モナークモナーク　●ロードモナークFirst

BIGLOBE
http://softplaza.biglobe.ne.jp/shop/falcom/

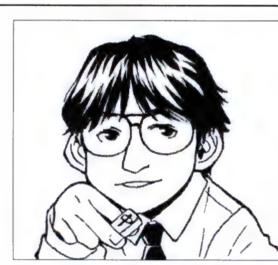
パピレス
http://www.papy.co.jp/

ファルコムの最新・最速情報をメールでGet！

毎月2回登録ユーザーに送られるファルコムのメールマガジン「Falcom News Express」をご存知ですか？どこよりも早い最新情報や，スタッフも知らない極秘情報をイチ早く知ることができますし，新製品の案内やホームページの更新情報などもお送りしています。登録はメールアドレスを入力するだけ，もちろん無料です。

これから新作も続々リリース予定ですので今すぐ登録して最新情報をGetしよう！

http://www.falcom.co.jp/wcgi/emregist.cgi



いよいよ次の新作の発売に向けての準備開始！ファルコムホームページでも，いくつかタイトルが紹介されています。どのタイトルが次に来るのか皆さんも気になるところでしょうが，まだ内緒です。突然の発表があるかもしれませんので，ファルコムホームページは常にチェックしておきましょう！



タイトル冠文：大阪府・御堂筋太郎
本文：暁

XRGB-2plus新発売！

なにわ通信

PRESENTED BY マイコンソフト

ADDRESS● 〒530-0005大阪市北区中之島3-2-4
朝日新聞ビル6F電波新聞社大阪本社内
マイコンソフト株式会社 なにわ通信係

http://www.micomsoft.co.jp

長らくお待たせいたしました。

1月末，ついに弊社の新製品「XRGB-2plus」が発売されました。

今回は，「XRGB-2plus」の特集をしてみようと思います。

おさらい

「XRGB-2plus」は，アップ・スキャンコンバーターと呼ばれる装置で，家庭用ゲーム機やDVDプレイヤーの映像などのNTSC映像を，パソコン専用ディスプレイに表示するための変換装置です。

ヘッドホン接続に対応

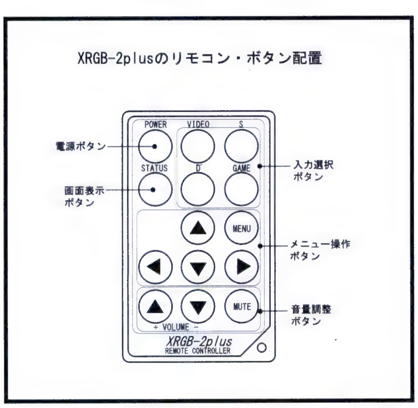
「XRGB-2plus」の音声出力端子は音量(ボリューム)調整ができるため，ステレオ・スピーカーだけでなくヘッドフォンにも対応できます。(音声出力端子は1系統のみですので，ステレオ・スピーカーとヘッドフォンを同時に接続することはできません。)

リモコンにも対応

「XRGB-2plus」はカード型リモコンが付属しています。

入力チャンネルをリモコンから変更したり，音量を調整することができます。

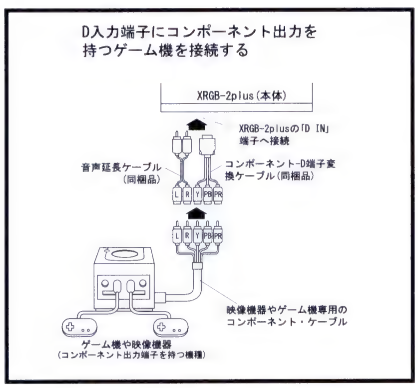
もちろん，各種機能も操作できるので，とても便利です。



コンポーネント・ビデオ入力対応

「XRGB-2plus」にはD入力端子が付いてり，任天堂さんの「GAME Cube」などD出力端子を持つゲーム機をD端子ケーブルで接続できます。

また，「XRGB-2plus」には，「コンポーネント-D変換ケーブル」が付属しており，コンポーネント・ビデオ出力端子を持つゲーム機(ソニーさんの「PlayStation2」や，マイクロソフトさんの「XBOX」)の映像を鮮明に入力することができます。

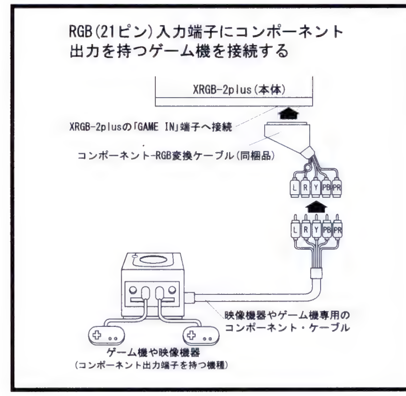


マルチ対応のRGB入力

「XRGB-2plus」には，RGB(21ピン)入力端子も付いており，ソニーさんの「PlayStation」や任天堂さんの「スーパーファミコン」など，旧型のRGB出力を持つゲーム機をRGB(21ピン)ケーブルで接続することができます。

また，「XRGB-2plus」には「コンポーネント-RGB変換ケーブル」が付属しており，RGB(21ピン)入力端子にコンポーネント・ビデオ出力端子を持つゲーム機の映像を入力することもできます。RGB映像入力を使用しない方には，コンポーネント・ビデオ入力端子として利用できるのでとても便利です。

(RGB信号/コンポーネント・ビデオ信号の入力切り替えは，XRGB-2plusの設定を変更する必要があります)



ガンマ補正と色選択機能

「XRGB-2plus」には，ガンマ補正機能が内蔵されており，画面を明るく表示することができます。通常のガンマ補正のほかに，R・G・B個別に微調整できる機能も付いています。旧型機の「XRGB-2」ではアナログ方式のガンマ補正機能でしたが，「XRGB-2plus」ではデジタル方式にすることによって，より細くガンマ値を設定できるようになりました。薄暗いダンジョンを旅するゲームなどで威力を発揮する機能です。

また，「XRGB-2plus」には，C_MODEという機能が付いており，ビデオ入力とS入力の映像の色合いを，通常よりも暖かく表示する機能が付いています。

パソコン専用モニターでは，どうしても色合いが冷たい感じに表示され，家庭用テレビと色合いを同じにすることは難しいのですが，C_MODE機能を使用することで，画面の肌色などを暖かく表示することができます。

目印は赤と青

「XRGB-2plus」のパッケージは，赤と青を基調としたハデなデザインになっています。

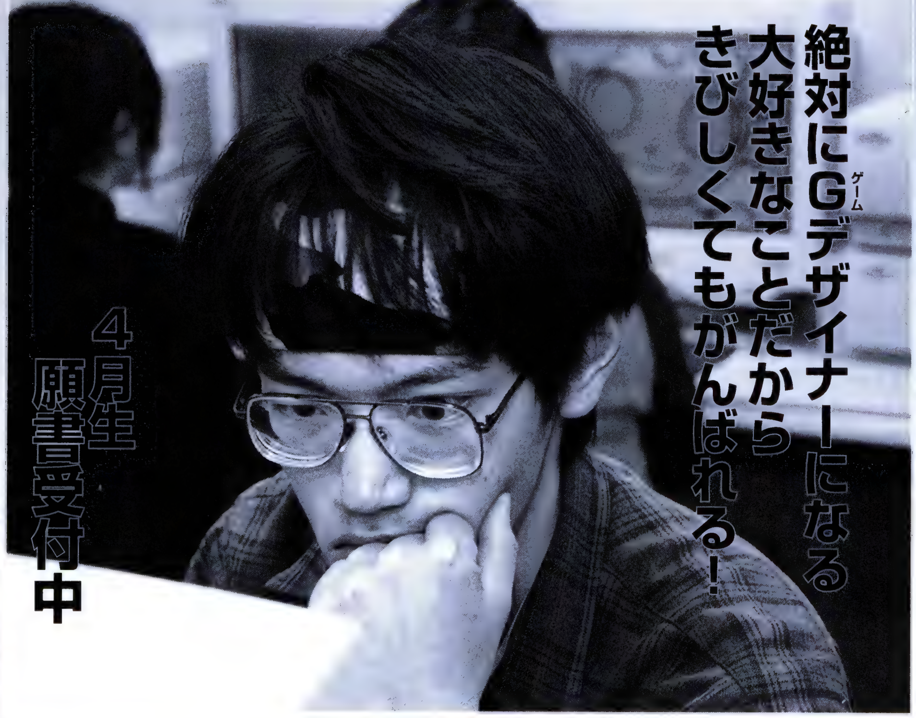
価格は28,800円(税別)となっております。

お近くのゲームショップやパソコンショップでご購入・お取り寄せいただけると思いますので，是非買って下さいね！

フライヤー募集 なにわ通信では，第4回のフライヤー(ペーパー)を募集いたします。今回は，特に枚数制限などは設けません。描きたい事を紙に表現していただき，なにわ通信までお送り下さい。締め切りは4月末です。

●ゲームデザイナー●プログラマー●グラフィッカー●キャラクターデザイナー●サウンドクリエーター

私達があなたをプロにします

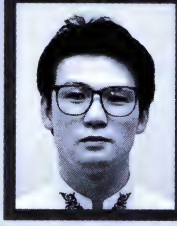


絶対にG(ゲーム)デザイナーになる
大好きなことだから
きびしくてもがんばれる!

4月生願書受付中

●全員集合!体験実習(無料)実施中 2月9日(土)・2月23日(土)・3月9日(土)申込みは下記へ!!

GAMEDESIGN TEACHER MESSAGE

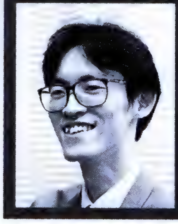


企画・仕様書作成とゲーム研究講座 担当の先生から

(株)日本テレネットに入社。その後、数々の会社で開発業務に携わる。現在は(有)ドリームスで開発管理室長と品質検査部事業部長を兼任。「この学院は、現役の先生が多く、学べることは無限です。企画作り・仕様書作りの一番新しい方法が学べます」

ゲームデザイン科担当講師 高橋一志

PROGRAM TEACHER MESSAGE



C言語プログラミング実習講座 担当の先生から

ソフト開発会社パーソナルビジネスアシスト社に6年間勤務し、プログラマーとしての実力を養う。「プログラムの知識がない人でも、基礎からみっちり教えます。実力がつく自主制作、ゲーム制作を多く行うカリキュラムが組まれています」

プログラム科担当講師 下村康幸

GRAPHIC TEACHER MESSAGE



グラフィックデザイン講座 担当の先生から

(株)日本テレネットで、ソフト開発ウルフチームに所属し「アークス1~3」「アイルロード」「天舞メガCD」「タイムギャル」「斬2」「夜叉演舞曲」等多くの作品を制作した。「この学科では、パソコンでの作画と紙面での作画を同等の割合で学べます」

グラフィック科担当講師 飯島公人

CHARACTERDESIGN TEACHER MESSAGE



キャラクターデザイン講座 担当の先生から

大手ゲーム制作会社を経て、自分が得意とするコンピューターグラフィック作家になる。コンピューターで描くイラストが好評で雑誌に連載した。「2D・3Dで描く、キャラクターデザインを指導しています。やはり、ゲームはキャラが命です」

キャラクターデザイン科担当講師 柏田幸子

DIGITALSOUND TEACHER MESSAGE



デジタルサウンド制作講座 担当の先生から

(株)キティ・エンタープライズソフト開発室等を経て、現在はゲームサウンド制作者。手掛けたゲームソフト「隠忍伝説」「ドラゴンボール2」等多数。「この学科はプログラムとDTMの両方を学習できるので、現場で必要とする技術が身につきます」

デジタルサウンド科担当講師 西澤 洋

今こんなゲームを創っています!



ここは現場と同じ

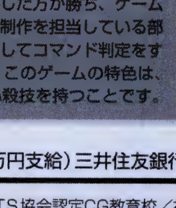
チームを組んでゲームを制作中

●タイトル・Knight Of Blow●ジャンル・対戦格闘●プレイ人数・1~2人●コンセプト・初心者や格闘ゲーム好きな人が楽しめるもの●ターゲット層・子供から格闘ゲーム愛好者まで幅広く。

●生徒チーム制作

Kinght Of Blow





プログラム科
戸高武将

開発中のゲーム内容
格闘ゲームが好きな人を中心に、誰でも楽しく遊べるようなシステムになっています。キャラクターを男1人、女2人の中から選び、先に相手の体力ゲージをゼロにした方が勝ち、ゲームクリアとなります。僕が制作を担当している部分は、キー入力処理、そしてコマンド判定をする処理をすることです。このゲームの特色は、キャラクターが4つの必殺技を持つことです。

■就職抜群!学生寮完備。朝日・毎日・読売新聞等奨学金制度(学費全額免除・給与月額12万円支給)三井住友銀行教育ローン有。学院見学入学相談随時可⇨日時をTELにて要予約

14年4月生願書受付中

ゲームデザイナー学部
全日制2年・定員300名

ゲームデザイン科
プログラム科
グラフィック科
キャラクターデザイン科
デジタルサウンド科



文部省認可CGARTS協会認定CG教育校/社団法人・日本漫画家協会賛助会員/社団法人・日本映画テレビ技術協会会員

日本ゲームソフト学院

〒101-0061 東京都千代田区三崎町3-4-9 ☎03-3261-8211

入学案内書 はがき(住所・氏名・年令・電話番号明記)か、電話で上記まで(無料)

FAX 0120-194-750 http://www.tag.co.jp/ E-mail:ngs@tag.co.jp

中古通販

PC NET ピーシーネット

在庫保有数日本最大級
(常時10,000台)

TEL.03-3733-1441

通販営業時間／月～金 9:00～19:00 土・日・祝日 11:00～19:30

●アドレスが新しくなりました
www.pc-net.jp/

今までのアドレスでも引き続きご利用いただけます。
www.prins.co.jp/

楽天市場店 http://www.rakuten.co.jp/pcnet/
ジャストネット http://emall.justnet.ne.jp/shop/pcnet/
ネットプライスモール http://www.netpricemall.com/
ビッグローブショッピングワールド http://shopping.biglobe.ne.jp/

最新情報は電話かネットでチェック!! 急げ!!

OS付きすぐ使える楽・楽パソコン!! PC NET イチオシ商品!!

三菱 NT4.0付	MICRON Win95付	富士通 Win95付	NEC Win95付
APRICOT LS550 (M3558-A1CO,A1NO) (MP233/32MB/3.2GHD/CD16/LAN) マニュアル付 U10000	SE 440BX (PenII(450)/128MB/8.4GHD/CD40/汚れ) マニュアル付 U10006	FMV 3CLH101,161,151 (PenIII(450)/64MB/6.4GHD/CD24/外装ヤケ) U10011	PC-9821 XC16/S5A3,S5B3 (Pentium(166)/32MB/1.6GHD/CD8/外装ヤケ/擦り傷) U1636
¥16,800	¥25,800	¥34,800	¥16,800

インターネットでは毎回お得な企画がいっぱい!!

3日間保証セール だから安い!!

期間限定タイムセール

全商品 外装クリーニング 無料! サービス

[安心の全品保証付] 1万円以上…1ヵ月保証 1万円以下…1週間保証

※本商品は添付品（OSを含む）は付属しておりません。
※中古パソコン販売時の在庫定義が変わりました。
今後在庫とは、従来品に加え、動作上支障のないキズ・天板やけ・カバー等の欠品がある物や、キーボード等が補充（他メーカー含）されている商品も、「在庫」としてお取り扱いさせていただきます。

DOS/V

注文No.	商品	価格
U8959	Aptiva(2190-22L)(IBM)(K6-2(450)/60MB/6.4GHD/CD40/外装焼け/省スペース型/PCIX3/USBX2)	¥26,800
U8344	Aptiva(2190-27N)(IBM)(K6-2(500)/60MB/10GHD/CD/台座無し/外装焼け/省スペース型/PCIX3/USBX2)	¥28,800
U1466	PC350(6587-JC3)(IBM)(MP(200)/32MB/4.2GHD/CD16/外装焼け/汚れ)……	¥6,900
U8920	PC300 GL(6561-27J)(IBM)(Celeron(266)/64MB/4.2GHD/CD32/外装焼け/擦り傷/PCIX2/ISAX1/共有X1/USBX2)	¥11,800
U1468	PC300 XL(6588-71J)(IBM)(PenII(266)/32MB/4GHD/CD16)……	¥13,800
U8708	PC300GL(6561-55,56J)(IBM)(PenII(300)/64MB/4.2GHD/CD32/PCIX2/ISAX1/共有X1/USBX2)	¥22,800
U1596	PC300PL(6871-12J)(IBM)(Celeron(400)/64MB/8GHD/CD40/省スペース型)	¥29,800
V0215	NETVISTA A40(6881-10J)◆Win98SE(IBM)(Celeron(533)/64MB/15GHD/CD24/100TX/省スペース型/付属品付)	¥45,800
U10057	PC-VC500J1FC1 D1(液晶セット)◆(送料2個口)(NEC)(PenIII(500)/128MB/13GHD/DVD(保証無)/15TFT(1024X768)/外装キズ/台座無/省スペース型)	¥59,800
U8779	PC-MA80T/C(液晶セット)◆送料2個口(NEC)(PenIII(800)/64MB/20GHD/CD40/15TFT/KB・マウス無/省スペースセットモデル)	¥79,800
U2700	PC-MA35D/C-63(NEC)(PenII(350)/64MB/6.4GHD/CD24/外装擦り傷/省スペース型/PCIX2/USBX2)	¥27,800
U8335	DESKPRO 2000/6300X(COMPAQ)(PenII(300)/64MB/3.2GHD/CD24/32MB増設/外装ヤケ/擦り傷/タワー型/PCIX2/ISAX2/共有X1/USBX2)	¥24,800
U8743	DESKPRO EP C266/3.2(COMPAQ)(Celeron(266)/64MB/3.2GHD/CD/PCカバー外装塗装剥げ/縦・横共用)	¥13,800
U1921	DESKPRO 6333/3.2(COMPAQ)(PenII(333)/64MB/3.2GHD/CD/PCIX2/ISAX2/USBX2)	¥26,800
U10019	DESKPRO EP-C400/6.4(COMPAQ)(Celeron(400)/64MB/6.4GHD/外装焼/擦り傷/フロントパネル爪折/PCIX2/ISAX1/共有X1/USBX2/AGPX1)	¥27,800
U8913	DESKPRO EP6400/6.4(COMPAQ)(PenII(400)/128MB/6.4GHD/CD/外装焼け/外装塗装剥げ/縦横共有)	¥32,800
U10063	OPTIPLEX GX1(DELL)(PenII(450)/128MB/10GHD/CD/マウス無し/汚れ/擦り傷/台座付/高さ:16.5cm/PCIX2/ISAX2/共有X1/USBX2)	¥35,800
U10064	OPTIPLEX GX1(DELL)(PenII(450)/256MB/8.4GHD/CD/マウス無し/汚れ/台座付/高さ:16.5cm/PCIX2/ISAX2/共有X1/USBX2)	¥37,800
U10027	OPTIPLEX GX100(DELL)(Celeron(500)/256MB/6.4GHD/CD/省スペース型)	¥39,800
U8941	Dimension J500C◆WinNT付(DELL)(Celeron(500)/256MB/13.6GHD/CD/省スペース型)	¥41,800
U10059	PRECISION 220(DELL)(PenIII(866)/256MB/10GHD/CD/マウス無し/汚れ/高さ:47cm/PCIX5/USBX2/AGPX1)	¥64,800
U10061	PRECISION 420(DELL)(PenIII(867)/512MB/9GHD/CD/高さ:47cm/PCIX5/USBX2/AGPX1)	¥69,800
U8881	GATEWAY 2000(G6-300)(GATEWAY)(PenII(300)/64MB/4GHD/CD24/外装焼け/タワー型(47cm)/PCIX3/ISAX1/共有X1/USBX2/AGPX1)	¥23,800
U10047	GATEWAY (GP7-450)(GATEWAY)(PenII(450)/256MB/10.1GHD/CD32/汚れ/外装キズ/外装焼け/高さ:48cm)	¥32,800
U10044	Vectra VL6/300-6(HP)(PenII(300)/64MB/6.4GHD/CD/外装ヤケ/汚れ/擦り傷)	¥22,800
V0144	ZHP-D6582N/801◆Win95付(HP)(PenII(400)/128MB/6.4GHD/CD32/100BT/電源ケーブル用変換コネクター無し/付属品付)	¥32,800
V0209	ZHP-D9456A/8J1◆Win98付(HP)(Celeron(500)/64MB/8.4GHD/CD24/100TX/スタンド無し/省スペース型/付属品付)	¥43,800
U10037	PANEL TOP LB40J16100(液晶一体型)(エプソン)(Celeron(400)/128MB/4.3GHD/CD/15TFT/外装焼け/KB型番違い/若干ドット抜け)	¥54,800
U1743	FMV62651,62653共通(富士通)(K6(266)/32MB/3.2GHD/CD24/外装キズ/外装焼け/スピーカー欠品/タワー型)	¥15,800
U1618	FMV-1DX7151(富士通)(PenII(266)/32MB/3.2GHD/CD24/外装焼け/擦り傷)	¥16,800
U8667	FMV-T73573M3(富士通)(PenII(350)/64MB/4.3GHD/CD32/640MO(保証対象外)/外装焼/スピーカ無/ミドルタワー型/PCIX3/ISAX1/共有X1/USBX2)	¥27,800

DOS/Vノート

注文No.	商品	価格
U20146	ThinkPad315D(2600-70J,7KJ)(IBM)(Pen(133)/48MB/1.08GHD/CD10/SVGA/ACアダプタ型番違い/KB文字剥げ/擦り傷/11.3TFT)	¥32,800
U6176	ThinkPad 365X(2625-7J6)(IBM)(Pen(100)/40MB/810HD/(32MB増設)/SVGA/擦り傷/AC型番違い/KB文字剥げ/10.4TFT)	¥11,000
U3756	ThinkPad 390(2626-CAJ)(IBM)(PenII(300)/64MB/3.2GHD/CD24/XGA/13.3TFT/USBX1)	¥73,800
U6120	ThinkPad 535(2606-MF9)(IBM)(Pen(133)/40MB/1GHD/SVGA/外装塗装剥げ/10.4TFT)	¥15,800
U6096	ThinkPad 560(2640-FJD)(IBM)(Pen(133)/8MB/1.08GHD/SVGA/AC型番違い/擦り傷/外装焼け/KB文字剥げ/12.1TFT)	¥14,800
U6056	ThinkPad 560X(2640-70J)(IBM)(MP(233)/32MB/4GHD/SVGA/AC型番違い/擦り傷/12.1TFT)	¥43,800
U20096	ThinkPad 760ED(9546-J3A,J31)(IBM)(Pen(133)/32MB/1.2GHD/CD4/SVGA/AC型番違い/擦り傷/外装塗装剥げ/12.1TFT)	¥26,800
U20062	PC-VP23C/WXAA1(NEC)(MP(233)/64MB/3GHD/CD20/XGA/外装焼け/擦り傷/13.3TFT)	¥46,800
U6394	ARMADA 7330T◆Win95付(COMPAQ)(MP(150)/32MB/2.1GHD/CD8/SVGA/外装焼け/PCカバー無し/ケース・マニュアル付/12.1TFT/16MB増設)	¥36,800
U6863	ARMADA 1590DMT(COMPAQ)(MP(166)/82MB/2.1GHD/CD/SVGA/擦り傷/ゴム足無し/12.1TFT)	¥39,800
U6274	FMV-5120NU2/W(FMV-52NU2W5.1共通)(富士通)(Pen(120)/32MB/1GHD/SVGA/擦り傷/KB文字剥げ/11.3TFT)	¥9,800
U6285	FMV-52NA2J1,5共通(FMV-5120NA2/W)(富士通)(Pen(120)/48MB/810HD/CD6/SVGA/外装擦り傷/11.3TFT)	¥15,800

注文No.	商品	価格
U6094	FMV-53NA2LS(富士通)(Pen(133)/16MB/1GHD/CD6/SVGA/ドット抜け/外装焼け/擦り傷/12.1TFT)	¥25,800
U6492	FMV-53NA3W1.5(FMV-5133NA3/W)(富士通)(Pen(133)/48MB/1.3GHD/CD10/SVGA/外装擦り傷/ヒンジ堅めの為異音有り/12.1TFT)	¥28,800
U6339	FMV-NU131[illegible](富士通)(Pen(133)/32MB/1.3GHD/CD20/モデムモジュラーケーブル無し/SVGA/[illegible]/12.1TFT)	¥26,800
U3605	FMV-5DNP5S(富士通)(Pen(133)/32MB/1.3GHD/CD8/SVGA/擦り傷/外装塗装剥げ/若干ピン折れ/12.1TFT/拡張ステーション)	¥28,800
U3575	FMV-55NA6W(FMV-5133NA6/W)(富士通)(MP(133)/32MB/1.6GHD/CD20/SVGA/[illegible]/12.1TFT)	¥29,800
U20109	FMV-55NA3W1.5(FMV-5150NA3/W)◆お買い得商品(現品のみ)(富士通)(Pen(150)/40MB/2.1GHD/CD10/SVGA/外装キズ/12.1TFT)	¥29,800
U6209	FMV-NF3161,2,3[illegible](富士通)(Pen(166)/32MB/2.1GHD/CD20/SVGA/バッテリー欠品/[illegible]/KB文字剥げ/12.1TFT)	¥29,800
U20127	FMV-NP3161,2,3(富士通)(MP(166)/64MB/2.1GHD/CD20/XGA/バッテリーパック欠品/12.1TFT)	¥29,800
U20006	FMV-3NU7XB0.5.6(富士通)(PenII(266)/32MB/3.2GHD/SVGA/擦り傷/開閉レバー不良/12.1TFT/USBX1)	¥32,800
U3948	DynaBook PA1230SA(東芝)(Pen(120)/32MB/1.35GHD/CD10/SVGA/外装擦り傷/外装焼け/KB文字剥げ(4)/11.3DSTN)	¥12,800
U6800	DynaBook PA1240S9(東芝)(Pen(133)/48MB/1.4GHD/SVGA/外装焼け/擦り傷/12.1DSTN)	¥12,800
U3326	DynaBook Satellite220(東芝)(Pen(133)/16MB/1.4GHD/SVGA/外装焼け/12.1DSTN)	¥11,000
U6577	PAP302JA.B(東芝)(MP(300)/64MB/4.3GHD/SVGA/10.4TFT/USBX1)	¥72,800
U20142	PC-5NU02-ZD7LA(日立)(MP(200)/64MB/3.2GHD/CD20/XGA/外装塗装剥げ/ゴム足無し/12.1TFT/拡張ステーション付(PC-BE5110))	¥36,800
U6327	AL-N1T515J5(パナソニック)(Pen(150)/48MB/1.35GHD/擦り傷/10.4TFT)	¥25,800
U6147	MEBIUS MN-5100(シャープ)(Pen(133)/32MB/1.4GHD/CD10/SVGA/擦り傷/11.3TFT)	¥21,800
U20102	MEBIUS MN-5410K(シャープ)(Pen(133)/64MB/1.6GHD/CD10/SVGA/擦り傷/11.3TFT/USBX1)	¥24,800
U6352	MEBIUS MN-6550D(シャープ)(Pen(150)/32MB/1.44GHD/CD10/モデム/XGA/CD脱着式/外装塗装剥げ/12.1TFT)	¥32,800
V0204	MEBIUS PC-PJ2-X5◆Win98付(シャープ)(PenII(366)/128MB/12GHD/XGA/56K/キーボード文字剥げ(2)/11.3TFT/B5サイズ/付属品付)	¥86,800
V0153	MEBIUS PC-FJ30◆Win98付(シャープ)(Celeron(333)/64MB/6.4GHD/CD20/トップカバー擦り傷/12.1TFT/付属品付)	¥69,800

バッテリーパックは消耗品につき保証対象外とさせていただきます。

98デスクトップ

注文No.	商品	価格
U2230	PC-9801BA3/U2(486DX2(66)/3.6MB/3.5×2/外装焼け/欠け有り)………	¥26,800
SH799	PC-9821BP/U7W◆Win3.1付(486DX2(66)/5.6MB/170HD/付属品付)….	¥14,800
SH800	PC-9821BP/U8◆Win3.1付(486DX2(66)/5.6MB/340HD/付属品付)……	¥19,800
U0241	PC-9821AS/U2(486DX(33)/3.6MB/FM/PCM/外装焼け/マウス無し/FDD部分若干欠け有り/2FDD)	¥29,800
U2129	PC-9821XN/U8(Pen(90)/7.6MB/270HD/PCM/2FDD仕様/外装焼け/擦り傷/スロットカバー無し)	¥19,800
U2631	PC-9821V233/M2(MP(233)/96MB/3GB/CD20/モデム無し/スピーカ・マイク無し)	¥34,800
SH057	PC-9821XA9/C8◆Win3.1付(Pen(90)/8MB/850HD/CD4/PCM/付属品付)	¥29,800
U8353	PC-9821XV20/W30(Pen(200)/96MB/3GHD/CD8/64MB増設/外装焼け/擦り傷/PCIX3/ミドルタワー型)	¥42,800
U8585	PC-9821RA223/W30(PenII(233)/96MB/3GHD/CD16/外装焼け/外装キズ/PCIX2)	¥36,800
U1584	PC-9821RA266/W30(PenII(266)/32MB/3GHD/CD16/外装焼け)………	¥36,800
U8788	PC-9821RA266/W・D30R(PenII(266)/128MB/3GHD/CD16/外装焼け/擦り傷/PCIX2/ISAX3)	¥44,800
U8558	PC-9821RA333/M60(Celelon(333)/32MB/6GHD/CD24/PCIX2)………	¥79,800

98ノート

注文No.	商品	価格
U3073	PC-9801NX/C120(486SX(20)/3.6MB/120HD/外装焼け/TFT)…………	¥17,800
U3420	PC-9821ND(8/340)(486SX(33)/8MB/340HD/KB焼け/9.5DSTN/メモリ・HD増設)	¥11,800
U3731	PC-9821ND2(16/500)(486DX2(50)/16MB/500HD/VGA/外装焼け/KB文字焼け/10.4DSTN)	¥14,800
U0333	PC-9821NE2(486SX(33)/1.6MB/9.5TFT)…………………………	¥12,800
U3097	PC-9821NE2/340(486SX(33)/5.6MB/340HD/外装焼け/ドット抜け(4)/擦り傷/9.5TFT)	¥14,800
U6273	PC-9821NE3/3(486DX2(50)/16MB/350HD/VGA/外装焼け/10.1TFT)….	¥15,800
U3293	PC-9821NE3/5(486DX2(50)/7.6MB/540HD/外装キズ/10.1TFT)………	¥14,300
U6872	PC-9821LA13/S14(Pen(133)/32MB/1.4GHD/擦り傷/外装塗装剥げ/10.4TFT)	¥19,800
U6704	PC-9821NA9/H8(Pen(90)/80MB/810HD/CD4/PCM/外装焼け/10.4TFT)‥	¥22,800
U50006	PC-9821NA12/H10(Pen(120)/32MB/1GHD/CD4/SVGA/10.4TFT)……	¥24,800
U6319	PC-9821NA12/S10F(Pen(120)/32MB/1GHD/CD4/SVGA/モデム無し/12.1TFT)	¥24,800
U6280	PC-9821NA13/H10(Pen(133)/32MB/2GHD/CD4/SVGA/12.1TFT)……	¥33,000
U6703	PC-9821NA13/H10(Pen(133)/80MB/1GHD/CD4/SVGA/12.1TFT)……	¥33,800
U3557	PC-9821NA15/X14(Pen(150)/16MB/1.4GHD/CD8/PCM/バッテリーパック欠品/12.1TFT)	¥34,800
U6382	PC-9821NA15/X14(Pen(150)/48MB/1.4GHD/CD8/PCM/擦り傷/12.1TFT/32MB増設)	¥42,800
U6760	PC-9821NR12/D10(Pen(120)/80MB/1GHD/VGA/擦り傷/12.1DSTN)….	¥17,800
U50001	PC-9821NR13/S14Z(Pen(133)/80MB/1.4GHD/SVGA/外装キズ/12.1TFT)	¥26,800
U6847	PC-9821NR15/S10(Pen(150)/16MB/3GHD/CD8/擦り傷/12.1TFT)……	¥32,800
U3852	PC-9821NR150/X14F(MP(150)/32MB/1.4GHD/CD10/モデム/XGA/12.1TFT)	¥47,800

バッテリーパックは消耗品につき保証対象外とさせていただきます。

MAC

注文No.	商品	価格
U9370	Performa 5430(PPC603e(160)/16MB/1.6GHD/CD8/15CRT(832X624)/外装焼け/一体型)	¥16,800
U9486	PowerMac 7600/200(PPC604e(200)/98MB/2GHD/CD12)…………	¥19,800
U9515	PowerMac 8500/150(PPC604(150)/80MB/1.2GHD/CD8/外装焼け)….	¥19,800
U9521	PowerMac 8500/180(PPC604e(180)/192MB/2GHD/CD8/外装焼け/擦り傷/ミドルタワー型)	¥21,800
U9380	PowerMac 8600/250(PPC604e(250)/160MB/4GHD/CD24/外装焼け/擦り傷/汚れ)	¥36,800
U9524	PowerMac 8600/250(PPC604e(250)/192MB/4GHD/CD24/ZIP・MO230付(保証無し)/外装焼け/擦り傷/ミドルタワー型)	¥39,800
U9518	PowerMac G3/DT233(PPCG3(233)/160MB/4GHD/CD24/外装焼け/外装キズ)	¥46,800
U9517	PowerMac G3/DT300(PPCG3(300)/160MB/6GHD/CD/ZIP(未チェックの為保証無し))	¥67,800
U9512	PowerMac G3-350(ブルー)(PPCG3(350)/128MB/6GHD/DVD/擦り傷/UBSマウス無し)	¥73,800
U9525	PowerMac G4/400(グラファイトカラー)(PPCG4(400)/320MB/20GHD/DVD(未チェックの為保証無し)/KB色違い/ミドルタワー型)	¥118,000

キーボードは別売となります。

ディスプレイ

注文No.	商品	価格
U4832	F14T41W-W(NEC)(14.1TFT:1024×768max/外装焼け/USB×1)……	¥29,800
U4496	PC-DV17A1(NEC)(17マルチ:1280×1024max/外装焼け/外装キズ/3列15pin)	¥10,800
U4896	E551ホワイト◆お勧め(新古品未開封)(DELL)(15マルチ:1024×768max/3列15pin)	¥13,800
U4897	E551グレー◆お勧め(DELL)(新古品未開封)(15マルチ:1024×768max/3列15pin)	¥13,800
U4889	E771ホワイト◆お勧め(新古品未開封)(DELL)(17マルチ:1280×1024max/3列15pin)	¥17,800
U4892	P1130ミッドナイトグレー◆お勧め(新古品未開封)(DELL)(21マルチ:1600×1200max/3列15pin)	¥59,800
U4890	P780グレー◆お勧め(新古品未開封)(DELL)(17マルチ:1600×1200max/3列15pin)	¥19,800
U4893	1503FP★今だけのお買い得商品!(新古品・未開封)(現品のみ)(DELL)(15TFT液晶:1024×768max/3列15pin)	¥39,800
U4891	P991グレー◆お勧め(新古品未開封)(DELL)(19マルチ:1600×1200max/3列15pin)	¥39,800
U4775	2236-3DN(IBM)(15マルチ:1280×1024max/外装焼け/3列15pin)………	¥4,900
U4728	LL-T155A(シャープ)(15TFT:1024×768max/外装キズ/3列15pin)………	¥35,800
U4277	GDM-17SE2T(SONY)(17マルチ:1280×1024max/外装焼け/擦り傷/3列15pin/5BNC)	¥12,800
U4411	CPD-20SF2(SONY)(20マルチ:1280×1024max/0.30mm/外装焼け/外装キズ/3列15pin)	¥22,800
U4798	RDT-141X(三菱)(14TFT:1024×768max/約26万色/3列15pin)………	¥31,000
U4666	RD15MII(三菱)(15マルチ:1024×768max/外装焼け/3列15pin)………	¥7,300
U4227	RD-17G2(三菱)(17マルチ:1600×1200max/0.25mm/外装焼け/外装キズ/3列15pin)	¥13,800
U4683	RDF17X(三菱)(17マルチ:1600×1200max/外装焼け/外装キズ/3列15pin)	¥14,800
U4146	MT-8617E(飯山電機)(17マルチ:1600×1200max/0.25mm/外装焼け/3列15pin/5BNC)	¥32,800
U4335	TXA3601GT(飯山電機)(14.1TFT:1024×768max/フルカラー/外装焼け/3列15pin)	¥32,800

プリンタ

注文No.	商品	価格
U0533	PC-PR700J(NEC)(136桁シリアル:漢字80字/秒/外装焼け/水平プリンタ)‥	¥59,800
U5128	LP-8300(エプソン)(レーザー:A3対応/外装焼け)………………	¥19,800
U5074	LP-9200(エプソン)(レーザー:A3対応/外装焼け/汚れ/フロントトレイカバー無し)	¥19,800
U5131	LP-9200SX(エプソン)(レーザー:A3対応/外装焼け)………………	¥42,800
U5156	MJ-5100C(エプソン)(カラーインクジェット:A3対応)……………	¥14,800
U5109	LBP-730(キヤノン)(レーザー:A3対応)…………………………	¥19,800
U5136	LBP-750(キヤノン)(レーザー:A3対応/外装焼け/外装汚れ)…………	¥21,800
U5092	LBP-830(キヤノン)(レーザー:A3対応)…………………………	¥22,800
U5171	LBP-850(キヤノン)(レーザー:A3対応/外装焼け/外装キズ)…………	¥29,800
U5241	XL-5800(富士通)(レーザー ハガキ～A3/600dpi/480dpi/28kg/デバイスドライバー付き/ユーティリティー付き/マニュアル付き'97年発売)	¥49,800

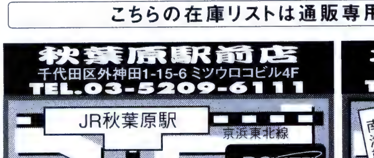
トナー等は消耗品につき標準添付されておりません。別途、ご確認下さい。

商品の状態により価格の変動がある事がございます。詳しくはTELかネット上でご確認ください。

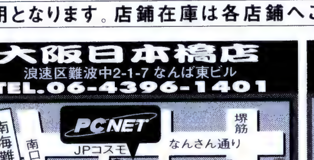
万一、品切れの際はご容赦ください。（1月25日現在）

こちらの在庫リストは通販専用となります。店舗在庫は各店舗へご来店の上ご確認ください。

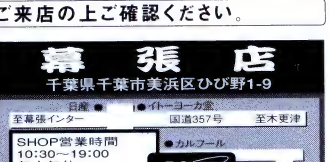
秋葉原駅前店
千代田区外神田1-15-6 ミツウロコビル4F
TEL.03-5209-6111



大阪日本橋店
浪速区難波中2-1-7 なんば東ビル
TEL.06-4396-1401



幕張店
千葉県千葉市美浜区ひび野1-9



パシフィック ネット
株式会社パシフィックネット
本社 〒144-0051 東京都大田区西蒲田7-10-1
相互西蒲田ビル TEL.03-3733-1441 FAX.03-3733-1440
通販営業時間／月～金 9:00～19:00 土・日・祝日 11:00～19:30

計測機器を各ショップのみで販売中。
詳しくは各ショップまでお問い合わせください。

iモードでショップ最新情報検索
http://www.prins.co.jp/i/

キリトリ線

月刊マイコンBASICマガジン③ 愛読者カード

★OFコーナー宛 ♥あくせすROOM宛 ■質問・感想（本誌への掲載／可・不可）

ペンネーム：

■アンケート回答記入欄■ ※黒の筆記用具でわかりやすく，はっきりとご記入ください

Q1		Q2	［メーカー名　］
Q3		Q4	Q5
Q6			
Q7		［商品名　］	
Q8			
Q9			
Q10		Q11	［④の人はソフト名を記入　］
Q12	［　］	Q13	［　］
Q14		Q15	

キリトリ線

■アンケートにご協力ください。

※左の愛読者カード・記入欄に，以下の質問に該当する番号をご記入ください。複数回答の質問で，回答欄をすべて埋める必要はありません。

Q1　メインで使っているパソコンは？
①デスクトップ ②ノート ③持っていない

Q2　パソコンの種類は？（Q1で③以外の方）
①自作 ②ショップブランド ③メーカー製
※枠内にメーカー名も記入してください

Q3　メインで使っているOSは？（Q1で③以外の方）
①Windows 95 ②Windows 98 ③Windows Me ④Windows 2000 ⑤Windows NT ⑥Windows XP ⑦Macintosh ⑧UNIX ⑨Linux ⑩その他

Q4　パソコン歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q5　インターネット歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q6　パソコンの主な使用目的は？（複数回答可）
①プログラミング ②ゲーム ③インターネット ④メール ⑤ミュージック ⑥学習 ⑦CG ⑧その他

Q7　これから買いたいと思うコンピュータ機器は？（複数回答可）
①パソコン本体 ②プリンタ ③スキャナ ④デジタルカメラ ⑤その他
※わかれば，商品名も記入してください

Q8　持っているゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥ゲームボーイ ⑦ゲームボーイカラー ⑧ゲームボーイアドバンス ⑨ワンダースワン ⑩ワンダースワンカラー ⑪その他

Q9　これから買いたいと思うゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥Xbox ⑦ゲームボーイカラー ⑧ワンダースワンカラー ⑨ゲームボーイアドバンス

※裏面へつづく→

■メモ欄…プログラム担当者・編集部へのご意見・ご感想など

Q10　携帯・PHSを持っていますか？
①両方持っている ②携帯 ③PHS ④購入予定 ⑤予定なし

Q11　ワープロ・ソフトは何をお使いですか？
①Word ②一太郎 ③メモ帳 ④その他

Q12　使用中のプログラミング言語は？
※枠内に一つご記入ください

Q13　これから勉強したいプログラミング言語は？
※枠内に一つご記入ください

Q14　今月号の本誌で面白かった記事を三つ，下の表からお答えください

Q15　今月号の本誌でつまらなかった記事を三つ，下の表からお答えください

ご協力ありがとうございました。

■今月号の記事一覧

キリトリ線

マイコンBASIC Magazine　プログラム投稿応募カード

言語名（音源名）					
プログラム名（曲名）					
氏　名		ペンネーム		年齢	歳
住　所	〒				
TEL	※連絡可能な時間帯		応募年月日	年　月　日	

キリトリ線

郵便はがき

50円切手をお貼り下さい

141-8715

（受取人）

東京都品川区東五反田1−11−15

電波新聞社

月刊 マイコンBASICマガジン 編集部

3月号 行

(フリガナ)		年齢	性別
ご氏名		歳	男・女
□□□-□□□□		電話	
ご住所	都・道 府・県		

職業：小学生／中学生／高校生／高専生／短大生／大学生／専門学校生／予備校生／アルバイト／公務員／会社員／その他（　　　）

本誌を 毎月・時々・初めて 読む　何人で読みますか（　）人

自分のホームページを　持っている・持つ予定・持っていない・予定なし

本誌のほかによく読む雑誌は？［　　　］

ベーマガお楽しみプレゼント③応募欄（126ページ掲載）	希望番号	品名
	番	


好評発売中
Windows 95/98/Me/2000/XP
DVD-ROM版
CD-ROM版
ー新価格版ー
ツヴァイ!!
Zwei!!
PCゲーム売上ランキング1位！
週刊アスキー「今週の売れ筋ソフト」(2001年12/17～12/30集計)、CD-ROM Fan「売れ筋ランキング」(2001年12月集計)
TECH Win「SALES RANKING」(2001年12/1～12/31集計)、ラオックス他販売店ランキング1位！(2001年12/20～12/26集計)
七瀬葵
描き下ろし
新イラスト
NEWパッケージで登場！
アクションファン、RPGファンに
全力でお勧めしておきたい名作である。
GAME WATCHより抜粋
(http://www.watch.impress.co.jp/game/docs/20011220/zwei2.htm)
アクションRPGの核といえる戦闘シーンは、ツヴァイは100点
満点といい切っていいだろう。痛快さを求める余り、ゲーム中盤
以降はアクションRPGというよりシューティングゲームに近い
印象だが、広いダンジョンを縦横に駆け抜けつつ、敵の集団とび
しばしやりあっていく感覚は、「イース」
にも通じた明快なおもしろさがある。
impress
GAME Watch
無料ダウンロード実施中!!
●ツヴァイ!! 店頭用デモムービー
●ツヴァイ!! スクリーンセーバー&壁紙
●オリジナル・サントラZwei!! 試聴版
●イース・ヒーリング 試聴版
試してZwei!!キャンペーン 今すぐ遊ぼう！
各ゲームwebサイトでZwei!!体験版を無料ダウンロード中！
※簡易マニュアルはファルコムホームページからダウンロードできます。
★Falcom News Express
ファルコムの最新・最速情報をメールでGet！登録は簡単アドレス入力のみ
詳細はこちら→ http://www.falcom.com/
ヤリ込めばヤリ込む程オモシロイ!!
攻略ガイド2/10発売。完全制覇を目指せ！
本編以外にも、色んな楽しみが満載！
▲Zwei!!シューティング
▲ペット育成
▲タイピングイース
■お求めは取扱い店またはカンタン便利なファルコム通販(falcom.comファルコムドットコム)もご利用下さい。■社員募集(新卒・既卒・中途)！詳しくはホームページ「会社案内・社員募集」をご覧ください。
Falcom®
NIHON FALCOM CORPORATION
〒190-0012 東京都立川市曙町1-14-13
☎042(527)6501 http://www.falcom.co.jp/



第二十一巻第三号通巻二三七号 昭和五十七年十二月六日第三種郵便物認可 二〇〇二年二月二十五日印刷 二〇〇二年三月一日発行
電波新聞社 〒141-8715 東京都品川区東五反田一－十一－十五 電話〇三(三四四五)六一一一

定価550円 本体524円 雑誌18361-03 T1118361030556

2002

わかる!動かせる!プログラムが組める雑誌

マイコンBASIC Magazine

3

集中連載!

数学①②

平成14年度大学入試センター試験出題

「BASIC言語」ズバリ解説

http://www.basicmagazine.net/

電波新聞社